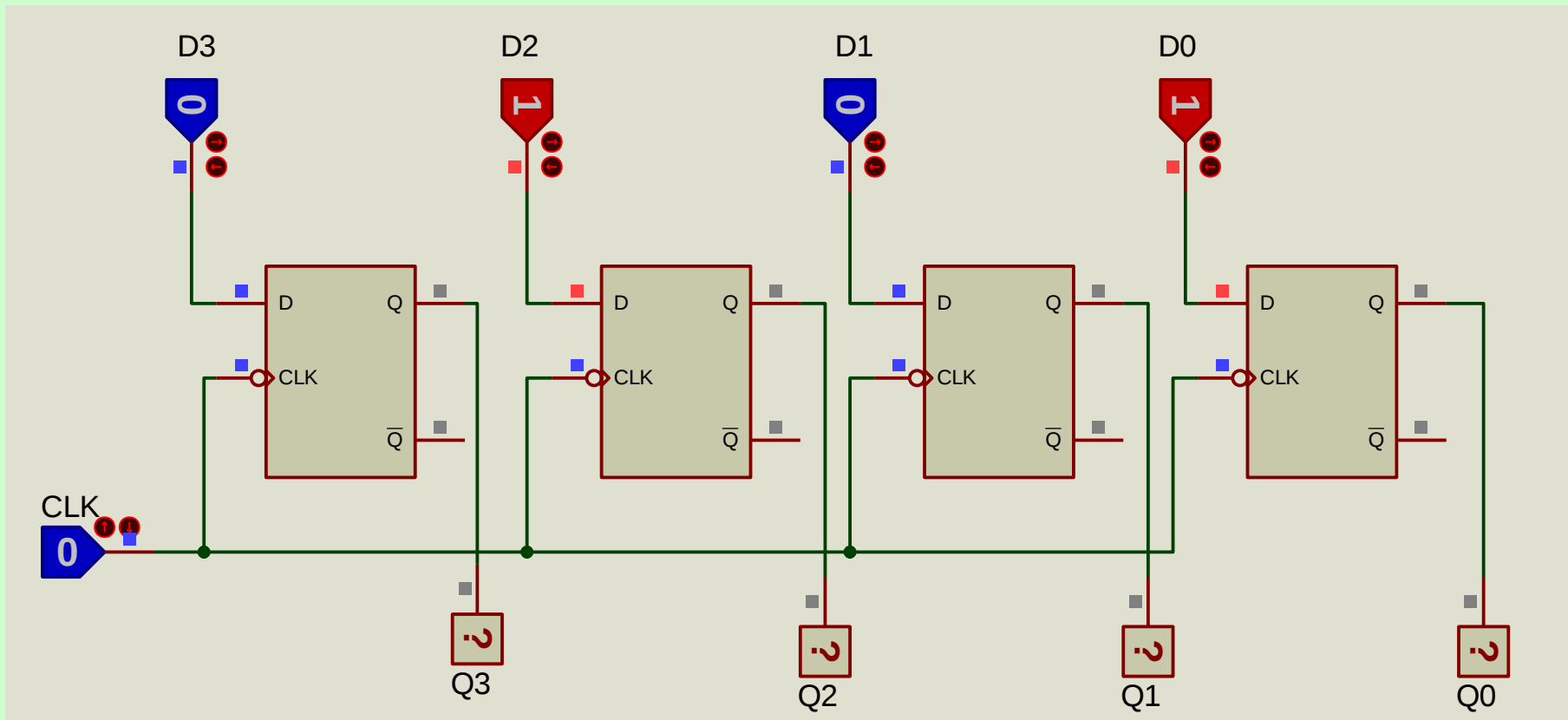

Introduction to Sequential Circuits Part 2

Registers

- Register is used for temporary data storage in all CPUs
- They are used for data storage based on FF
- So it consists of one or more FF
- Mainly there are two types of Registers
 - Parallel Registers
 - Serial Registers

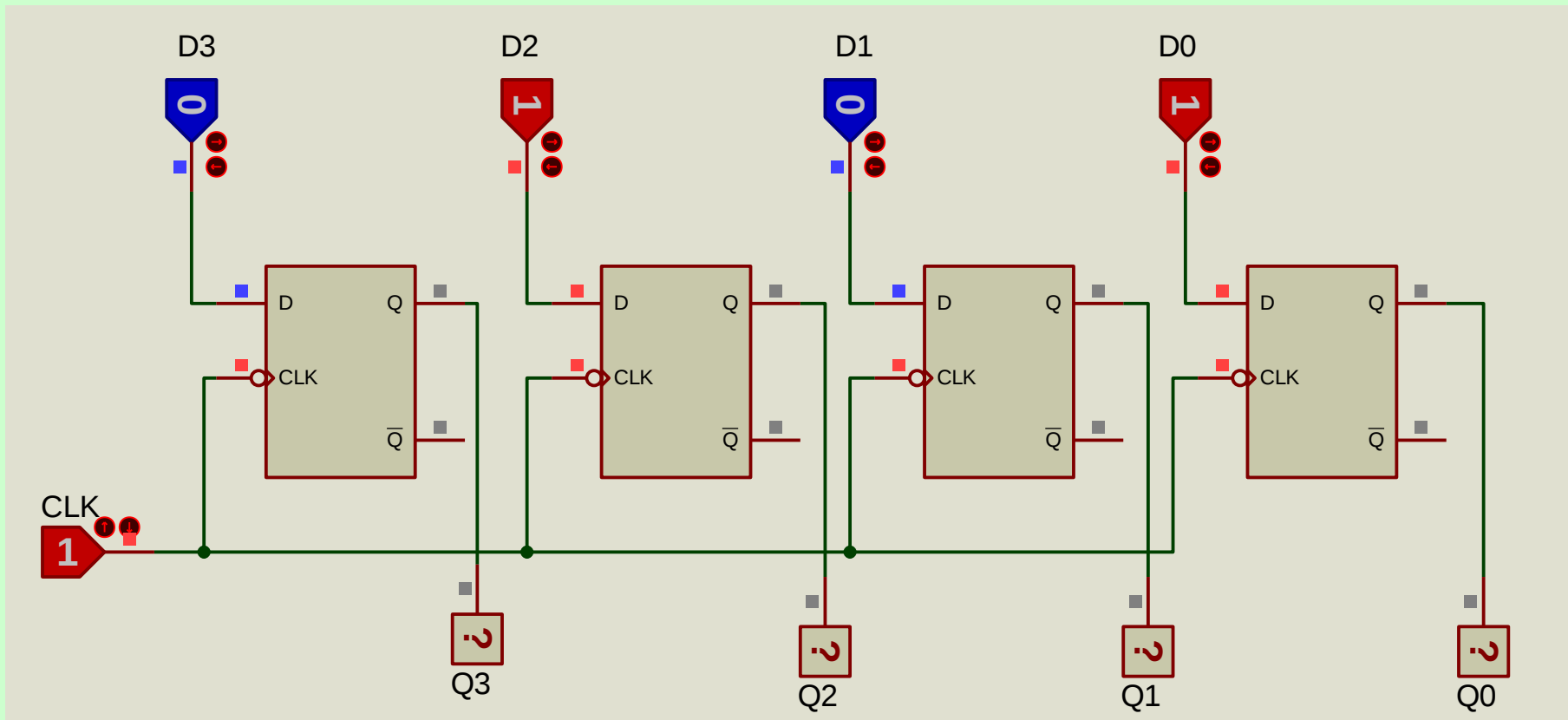
Parallel Registers

- Parallel Registers store all data bits at the same time
- For 4-bit parallel register it consists of 4-D type FF



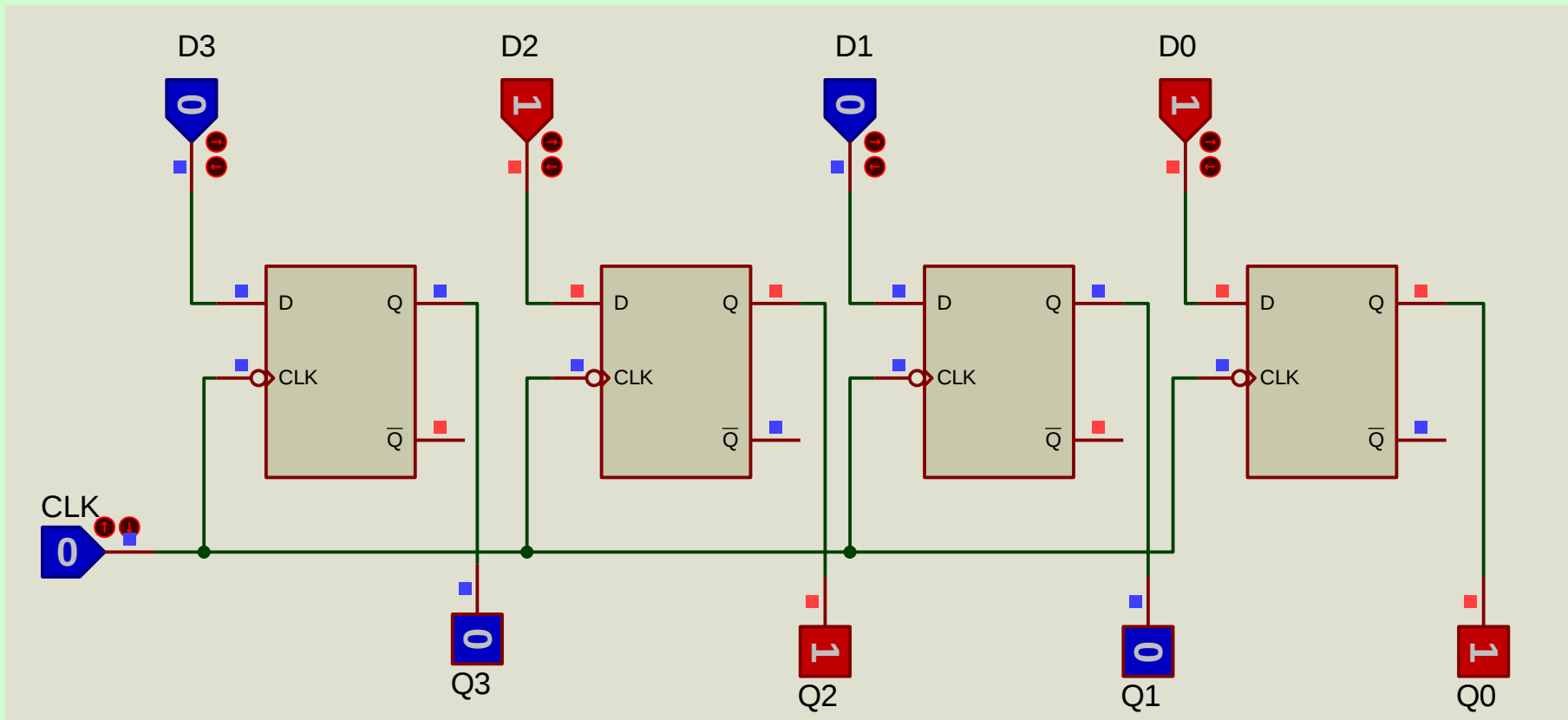
Parallel Registers

- Parallel Registers store all data bits at the same time
- For 4-bit parallel register it consists of 4-D type FF



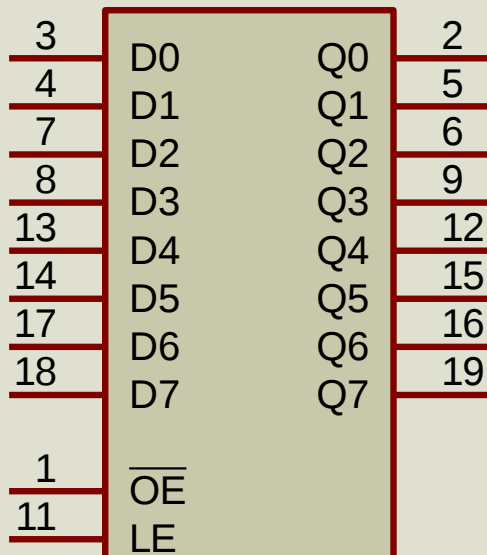
Parallel Registers

- Parallel Registers store all data bits at the same time
- For 4-bit parallel register it consists of 4-D type FF

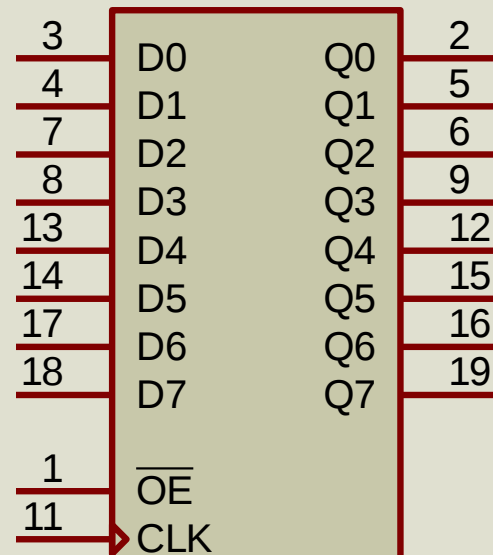


Parallel Registers Practical Examples

- 74LS373 → Octal transparent latch with 3-state output
- 74LS374 → Octal D-type FF with tri-state output



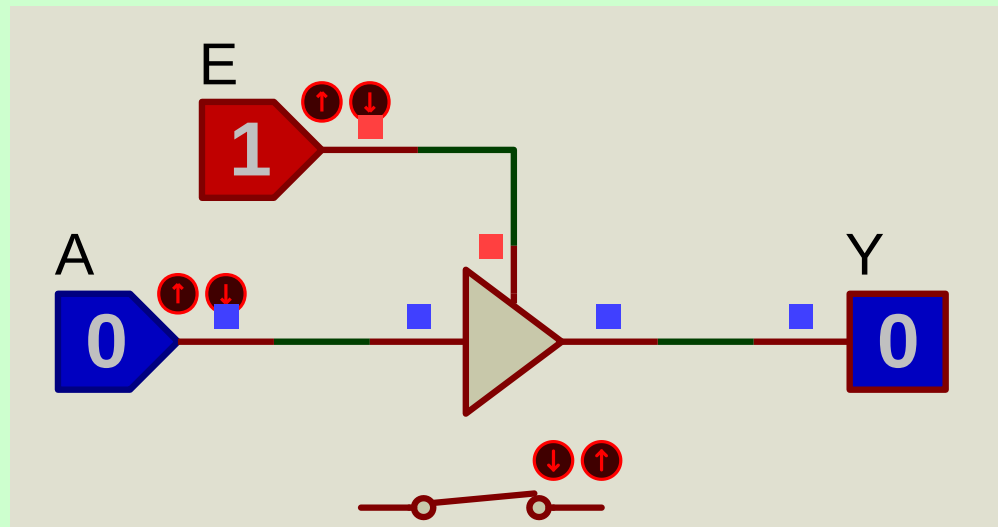
74LS373



74LS374

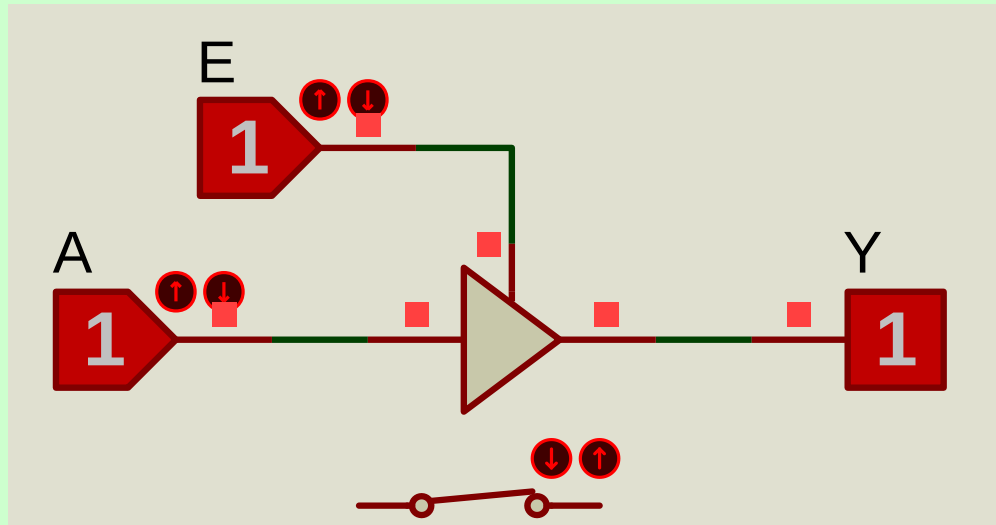
What is Tri-state or 3-State ?

- Logic circuits are two state system 0/1 (High-Low)
- 3-state means the system will have additional state to the two state system (High-Low – High impedance)
- A tri-state buffer replicates the input if $E = 1$



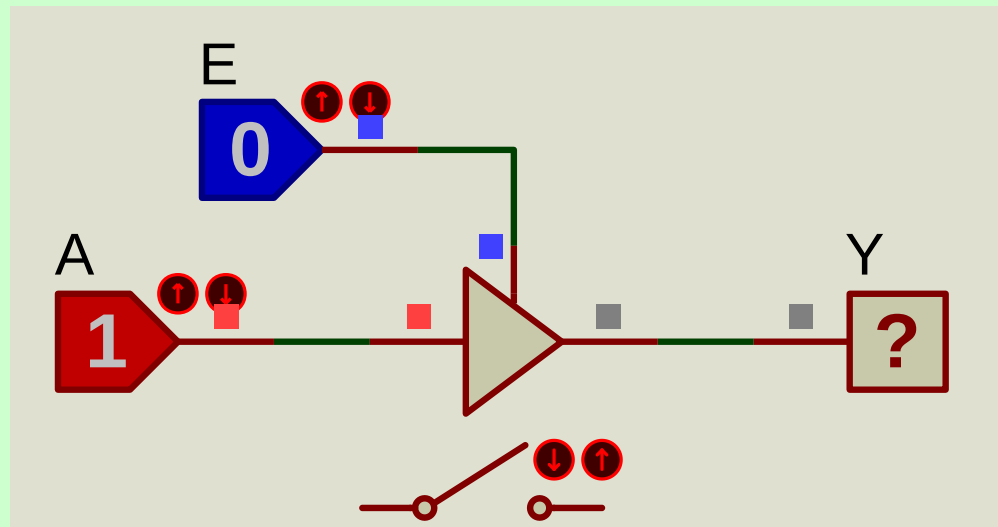
What is Tri-state or 3-State

- Logic circuits are two state system 0/1 (High-Low)
- 3-state means the system will have additional state to the two state system (High-Low – High impedance)
- A tri-state buffer replicates the input if $E = 1$
- If $E = 0$ the output of the buffer is disconnected or open circuited



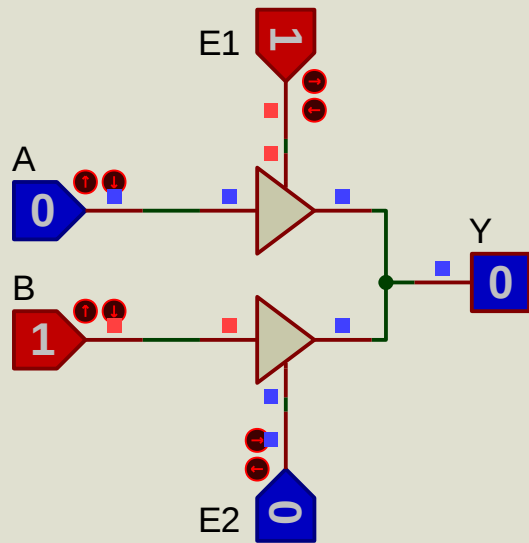
What is Tri-state or 3-State

- Logic circuits are two state system 0/1 (High-Low)
- 3-state means the system will have additional state to the two state system (High-Low – High impedance)
- A tri-state buffer replicates the input if $E = 1$
- If $E = 0$ the output of the buffer is disconnected or open circuited

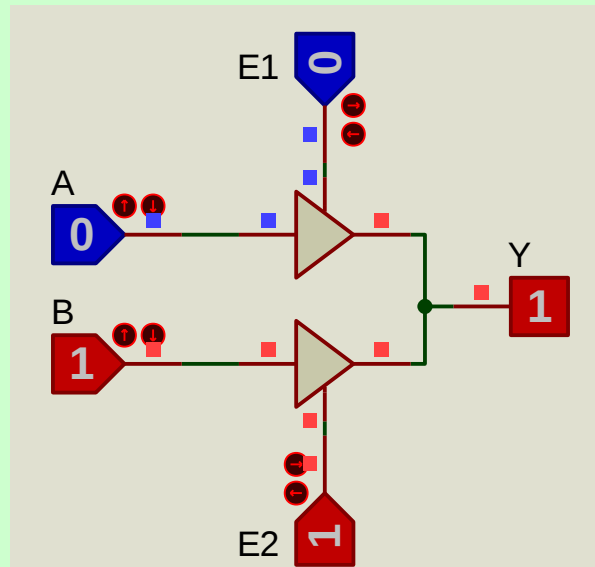


Tri-state buffer as MUX

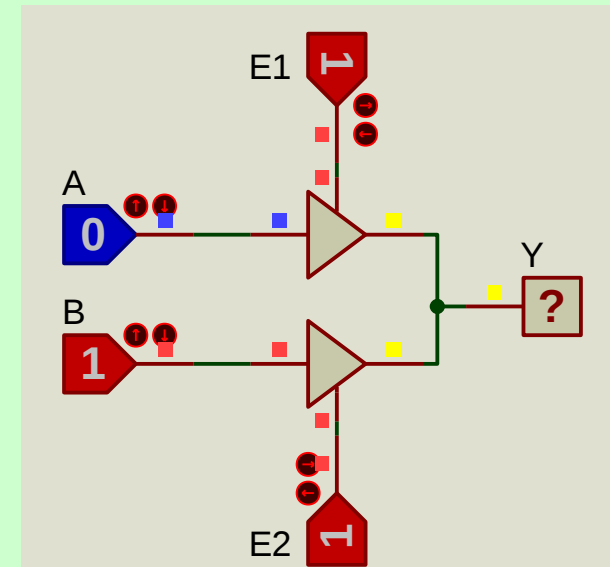
- Tri-state buffer can be used to implement data multiplexing (Multiplexer)
- Multiplexer is an essential part in the design of CPUs
- The figure shows how to use tri-state buffer as MUX



Select A

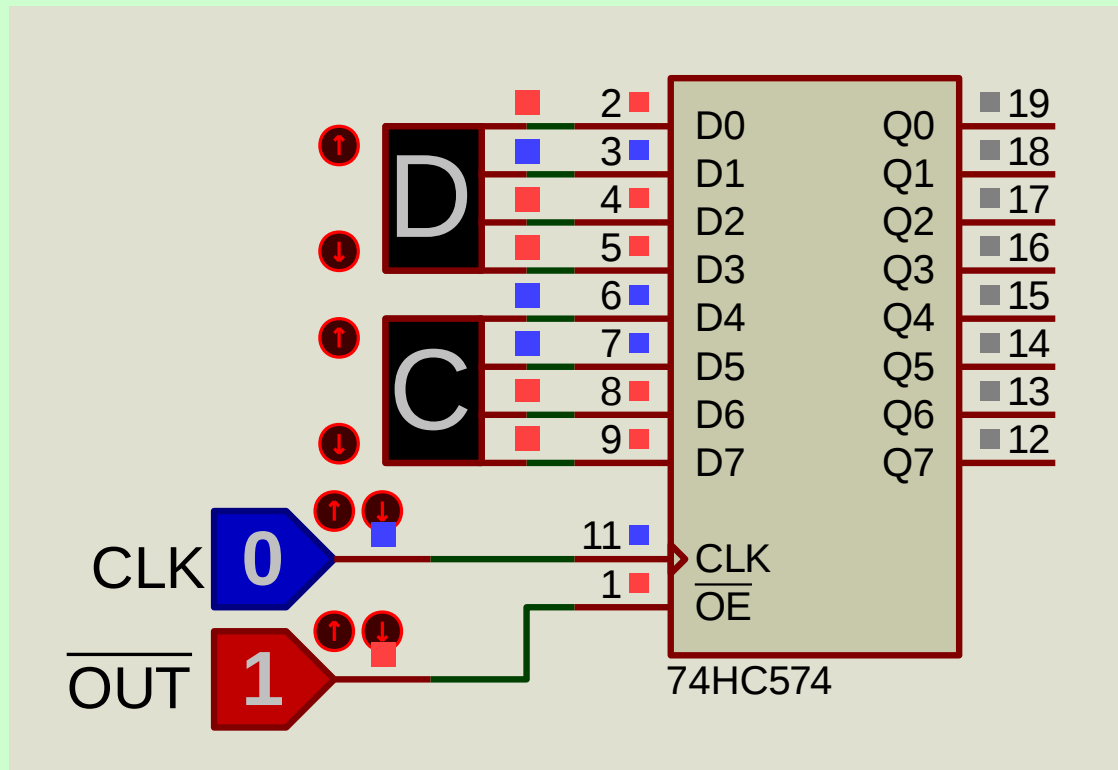


Select B

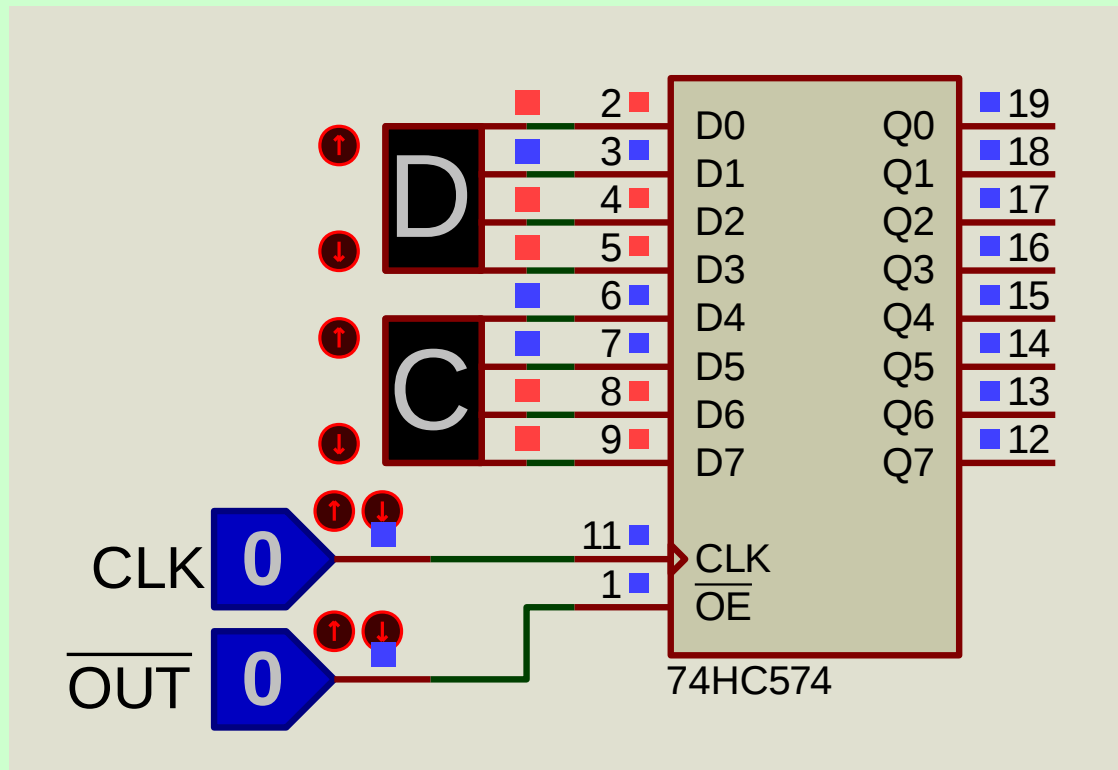


Select A,B

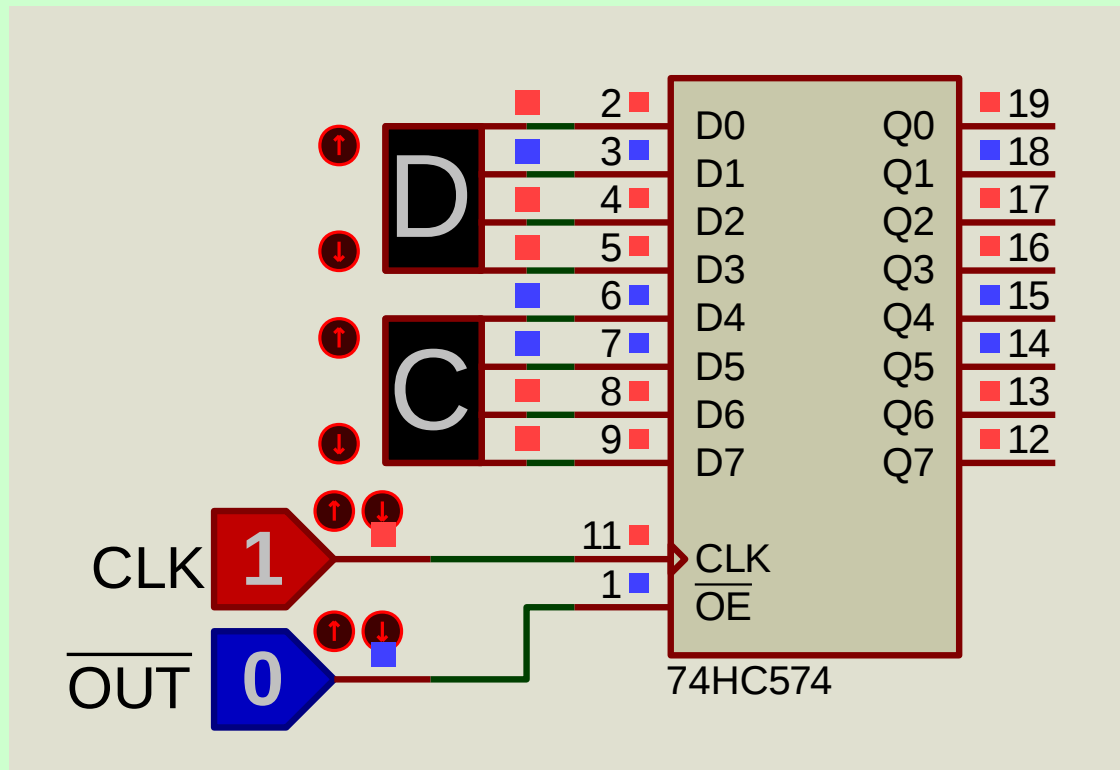
Typical Logic IC 74LS374



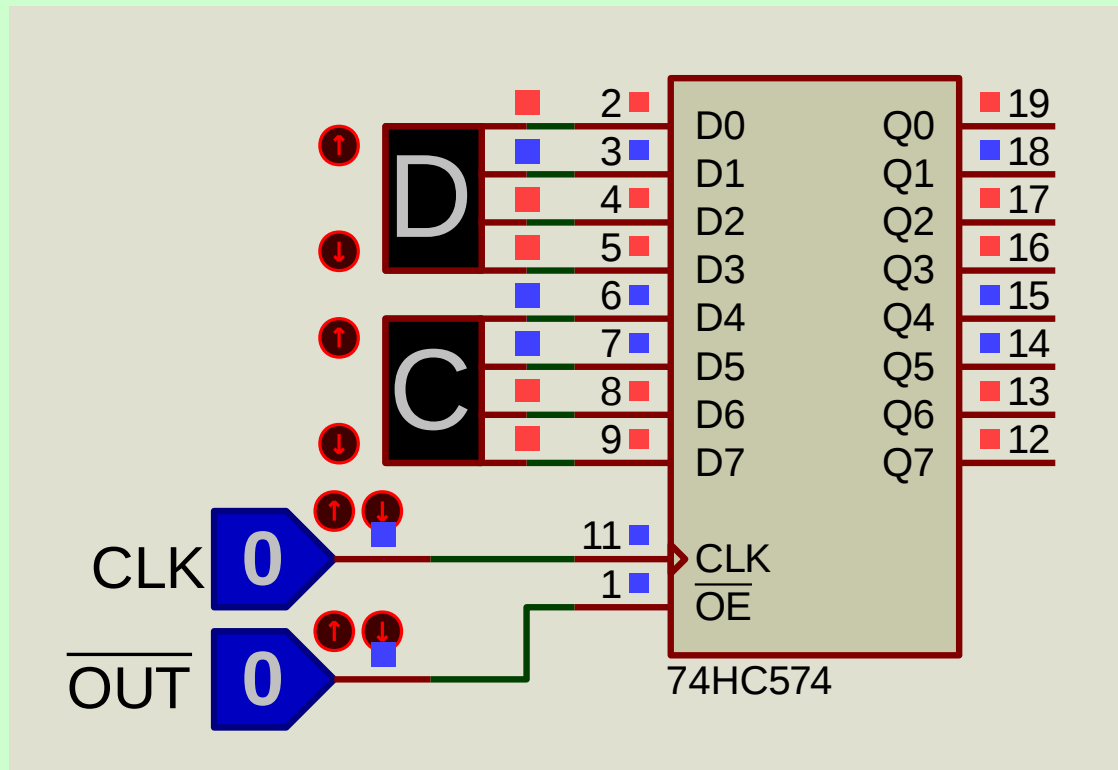
Typical Logic IC 74LS374



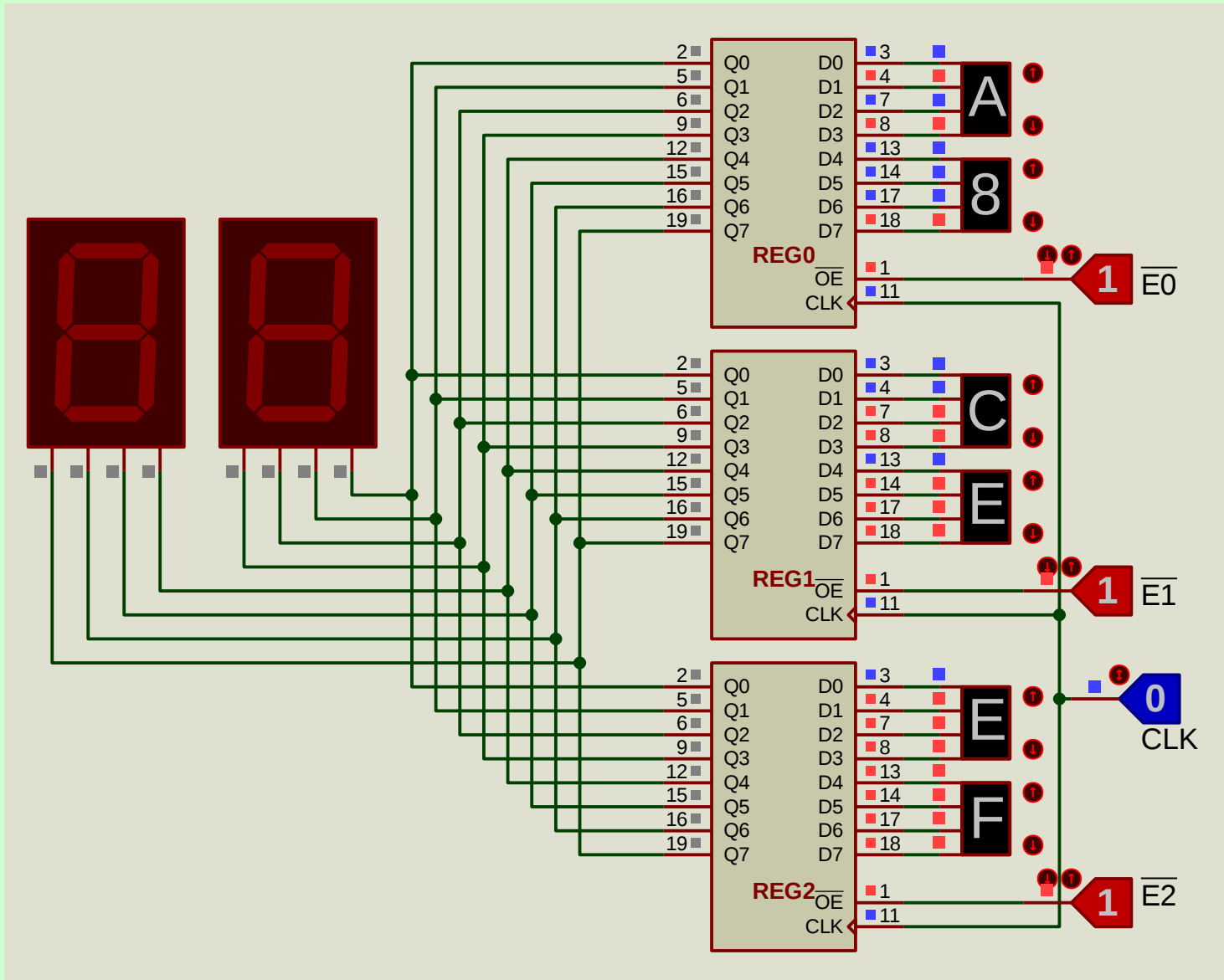
Typical Logic IC 74LS374



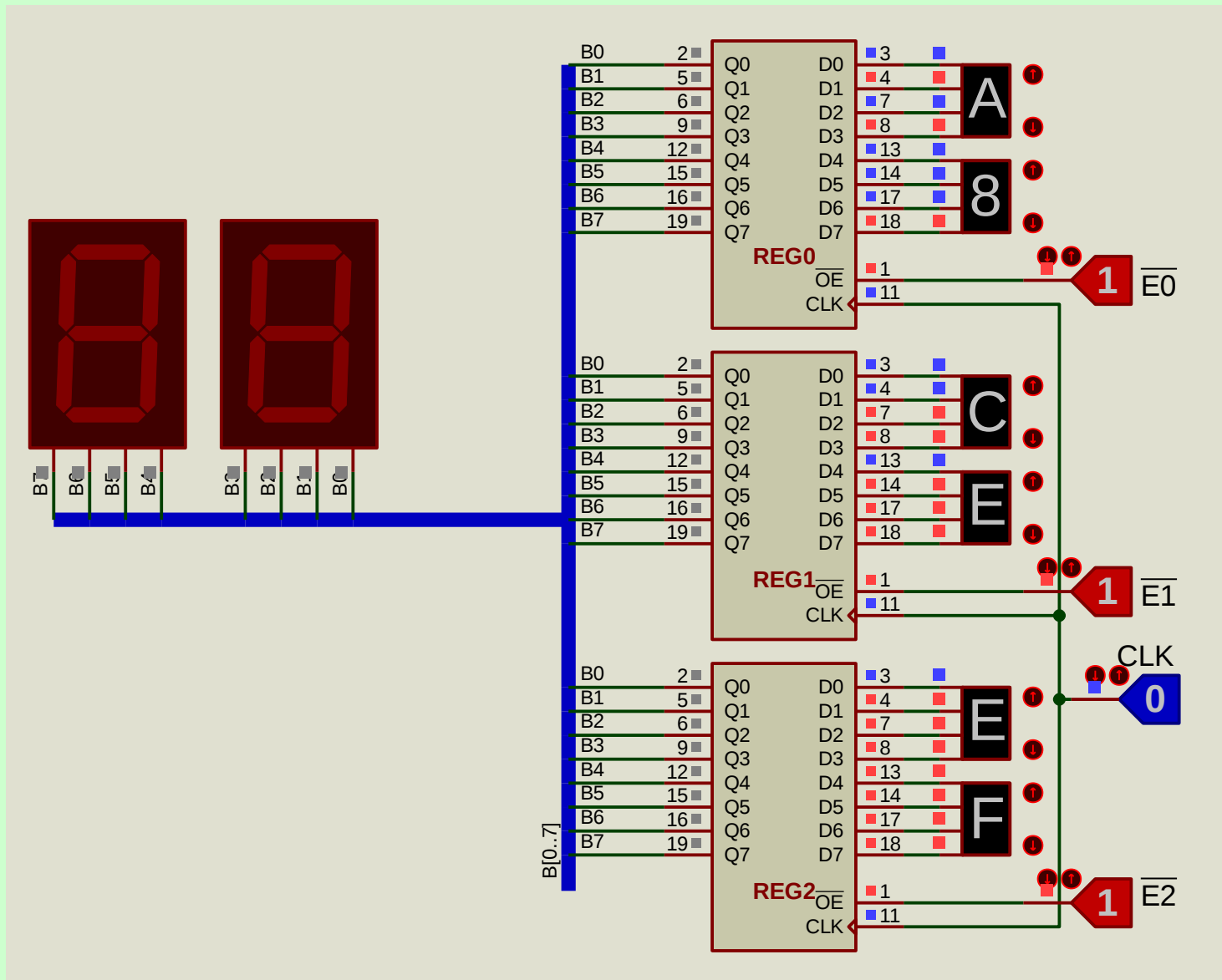
Typical Logic IC 74LS374



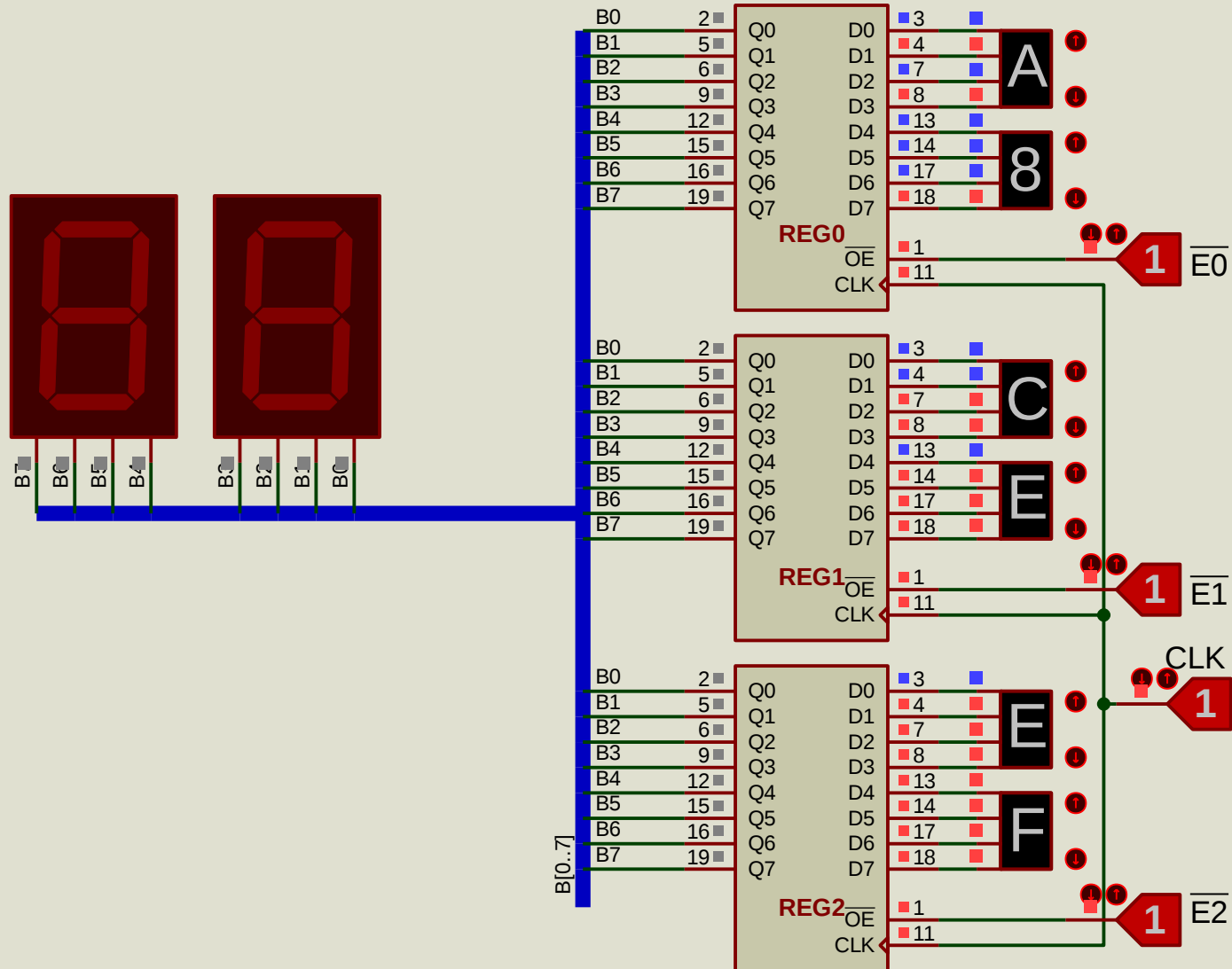
Bus MUX using tri-state buffers



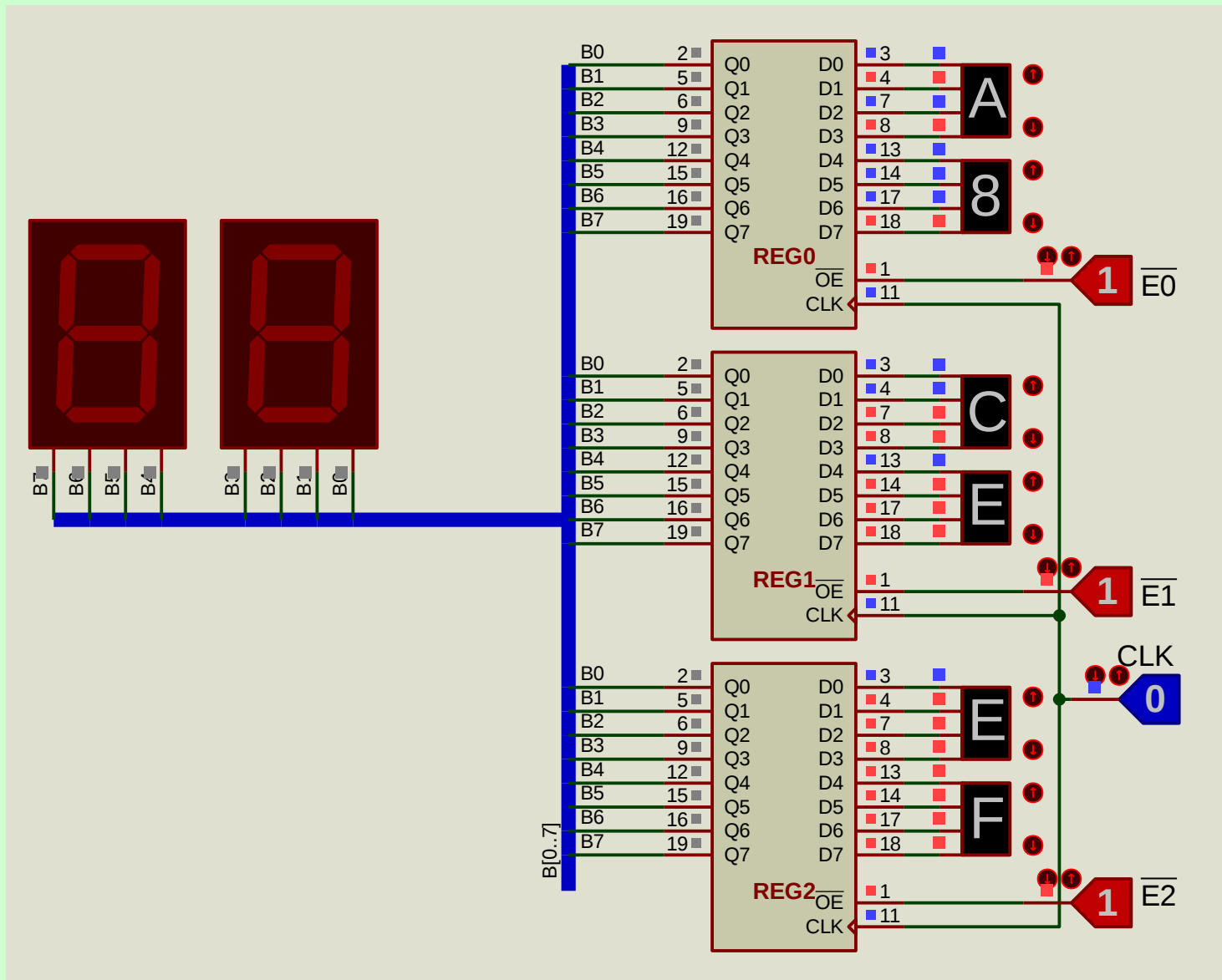
Bus MUX using tri-state buffers



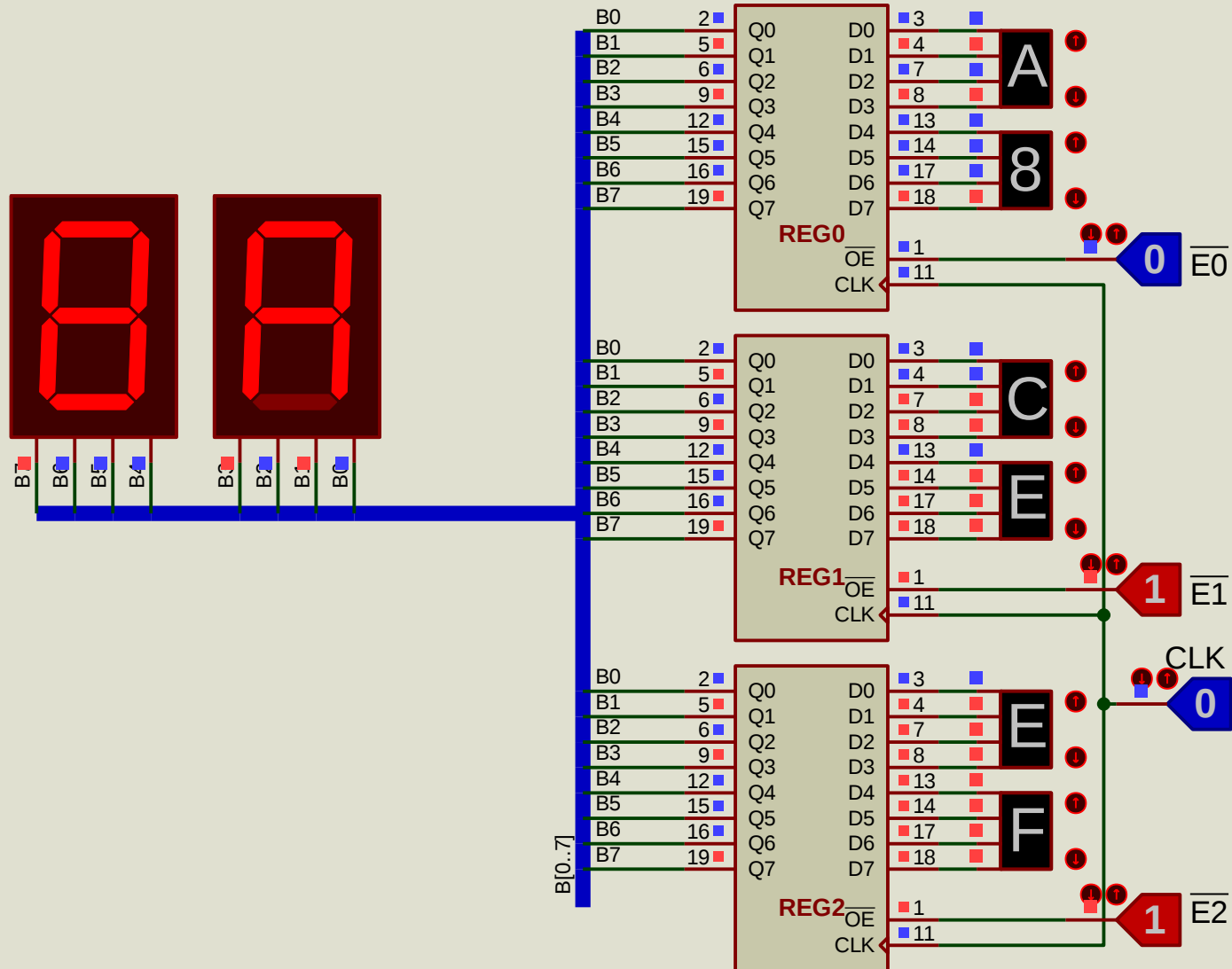
Bus MUX using tri-state buffers



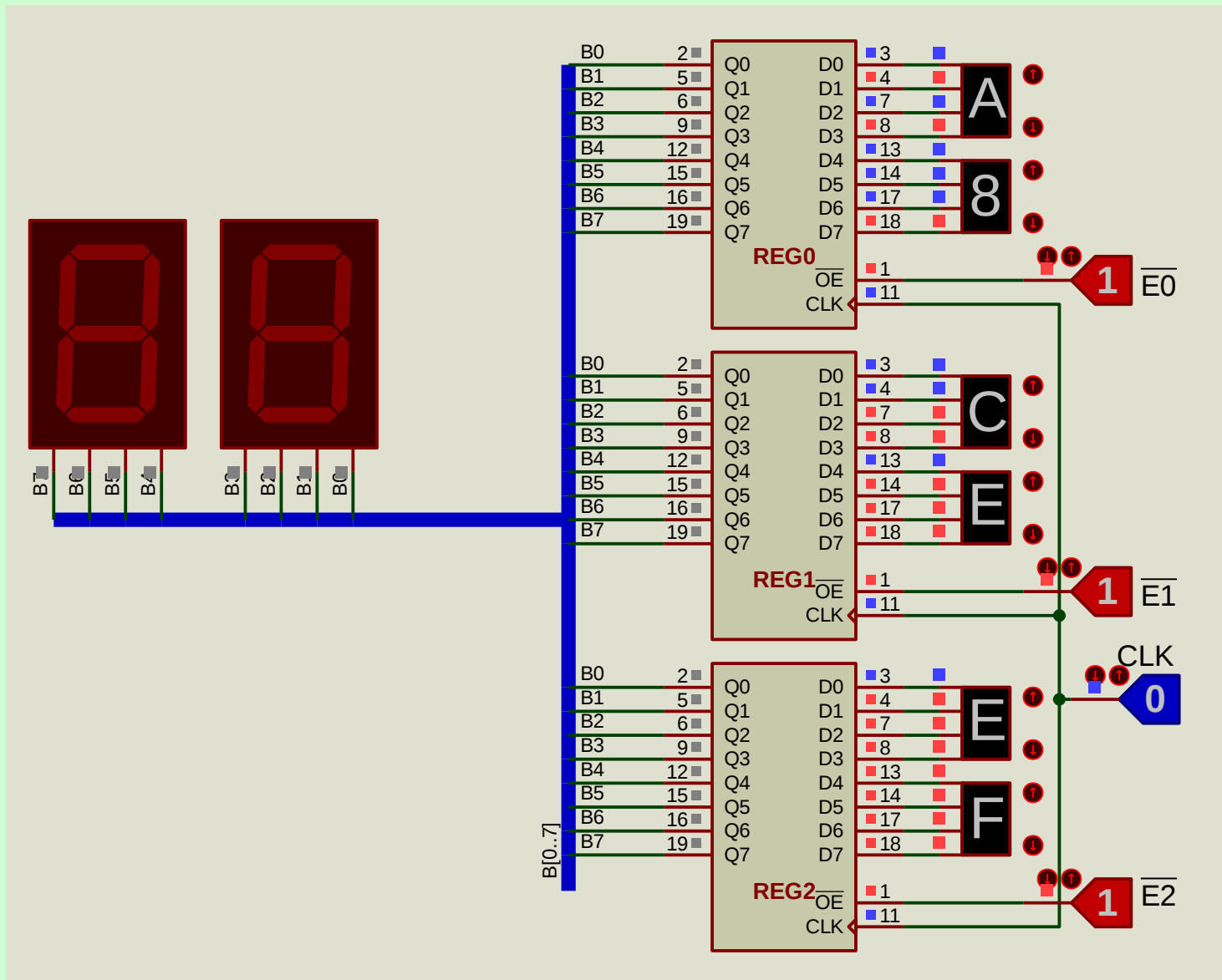
Bus MUX using tri-state buffers



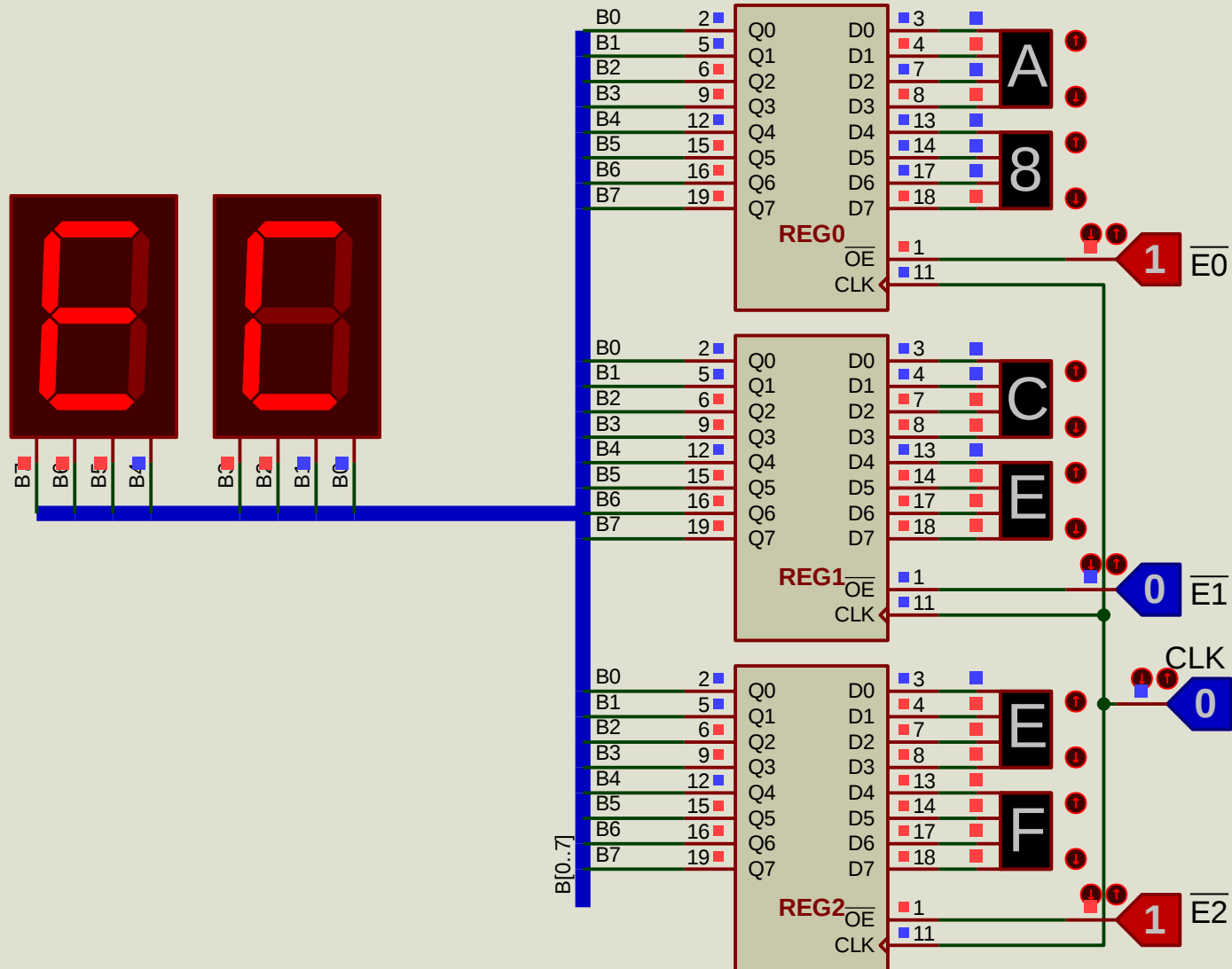
Bus MUX using tri-state buffers



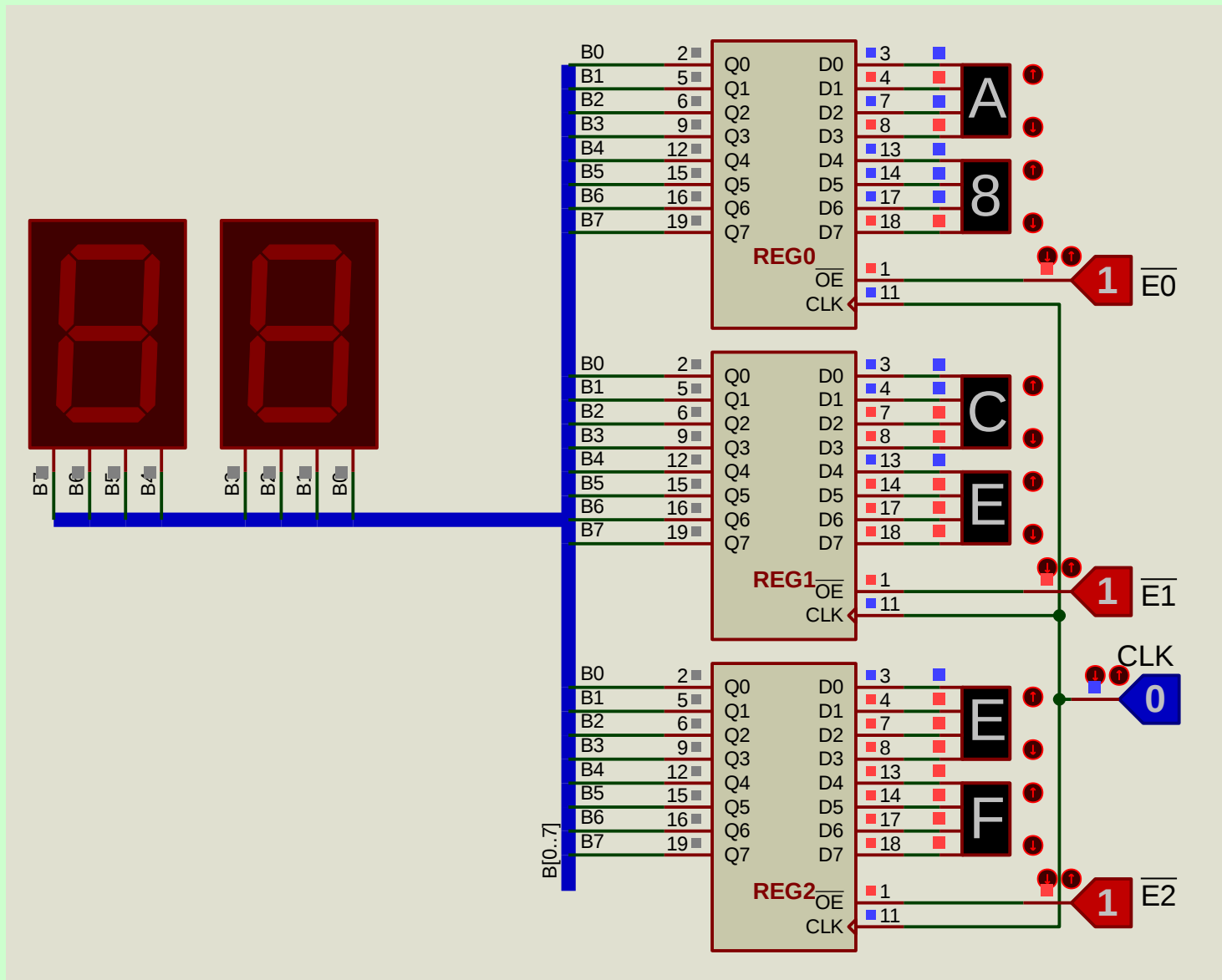
Bus MUX using tri-state buffers



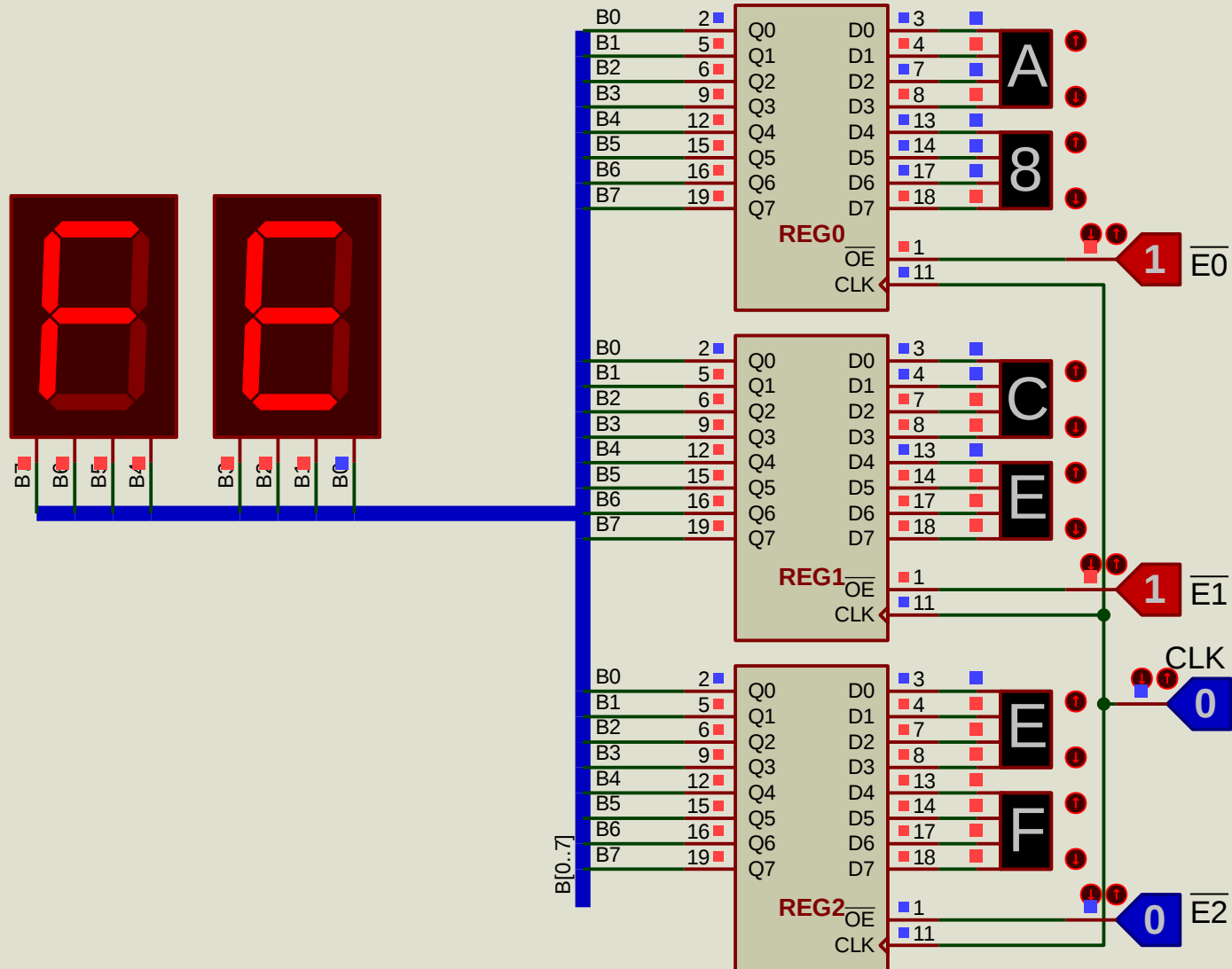
Bus MUX using tri-state buffers



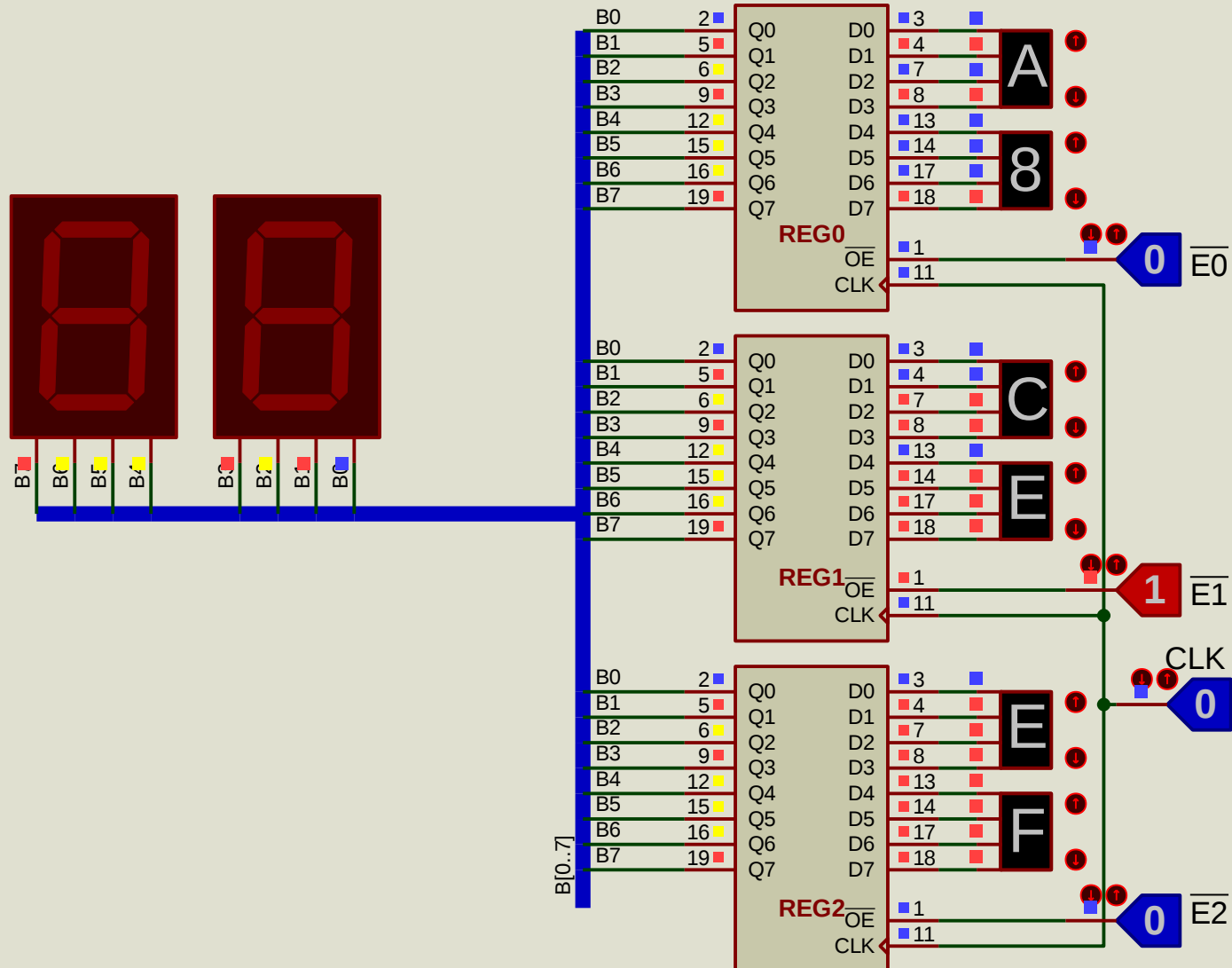
Bus MUX using tri-state buffers



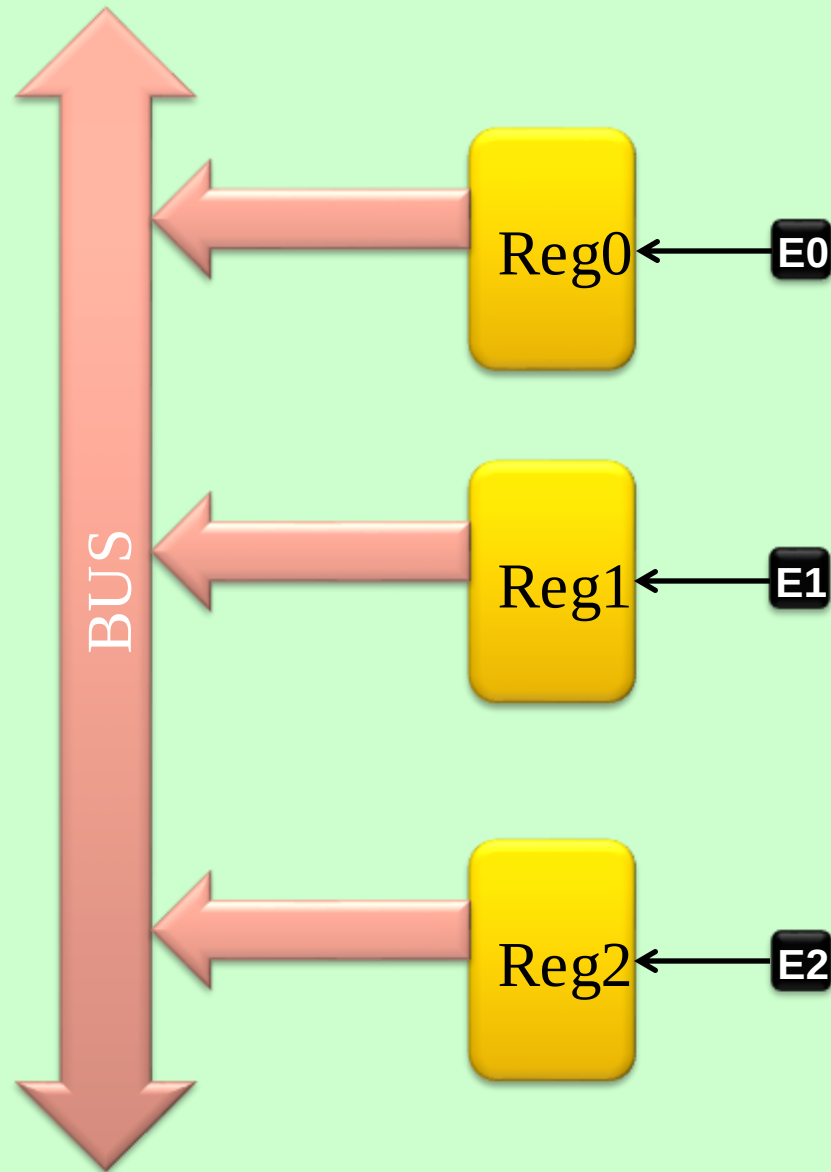
Bus MUX using tri-state buffers



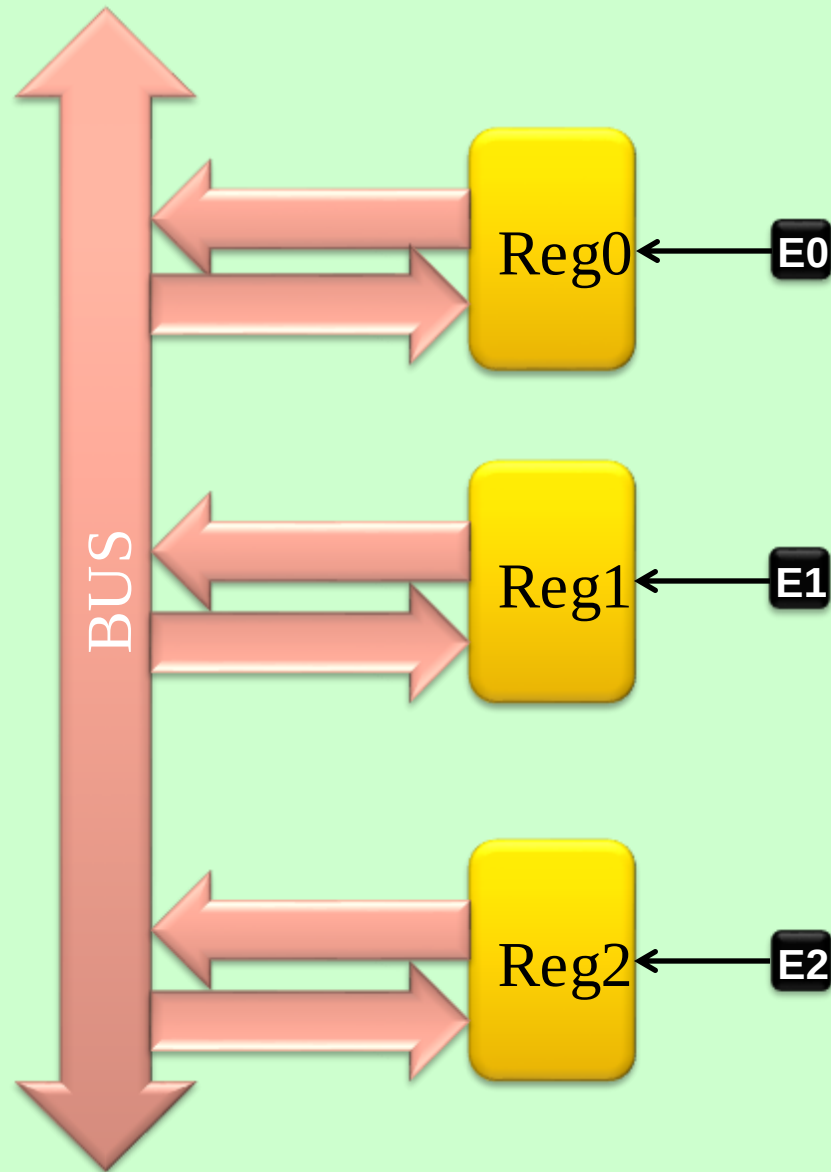
Bus MUX using tri-state buffers



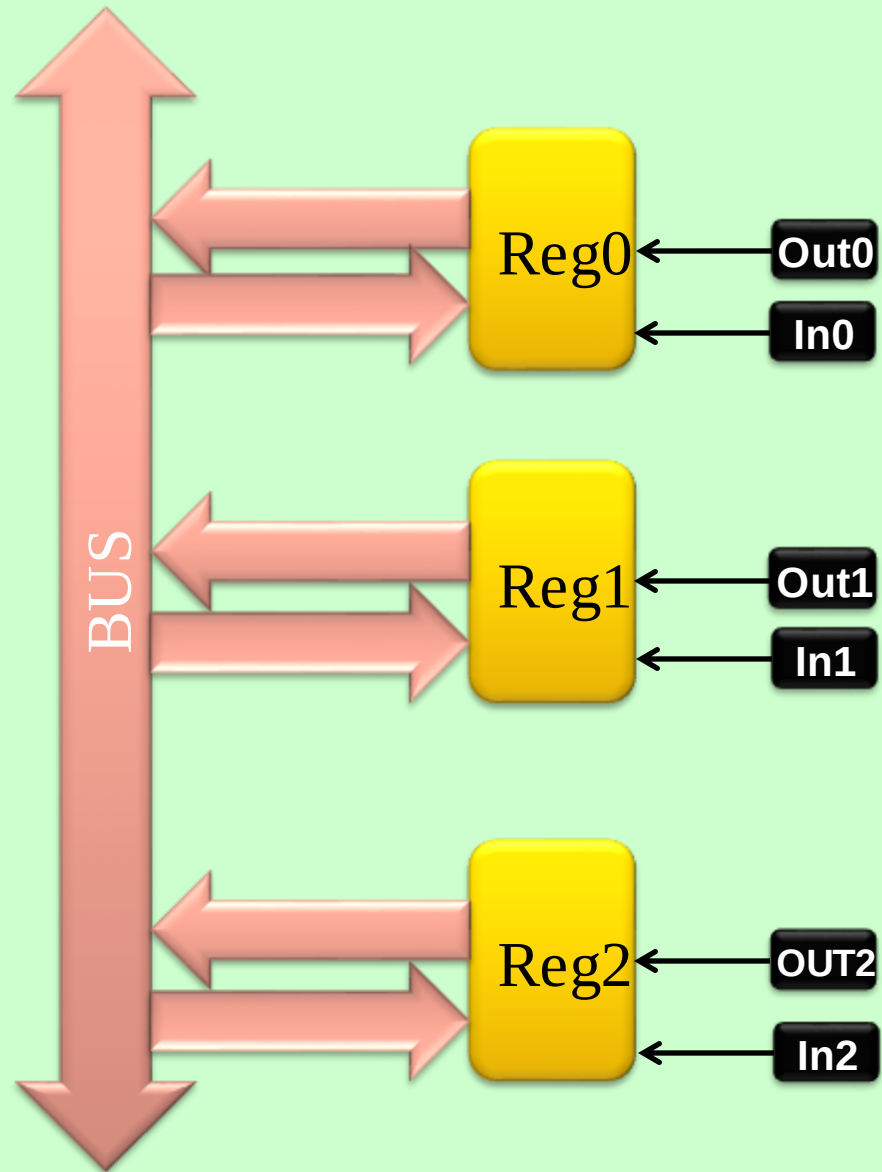
Registers to Bus connection



Registers to Bus connection



Registers to Bus connection



Registers to Bus connection

We need a bidirectional register that can be controlled to in or out data

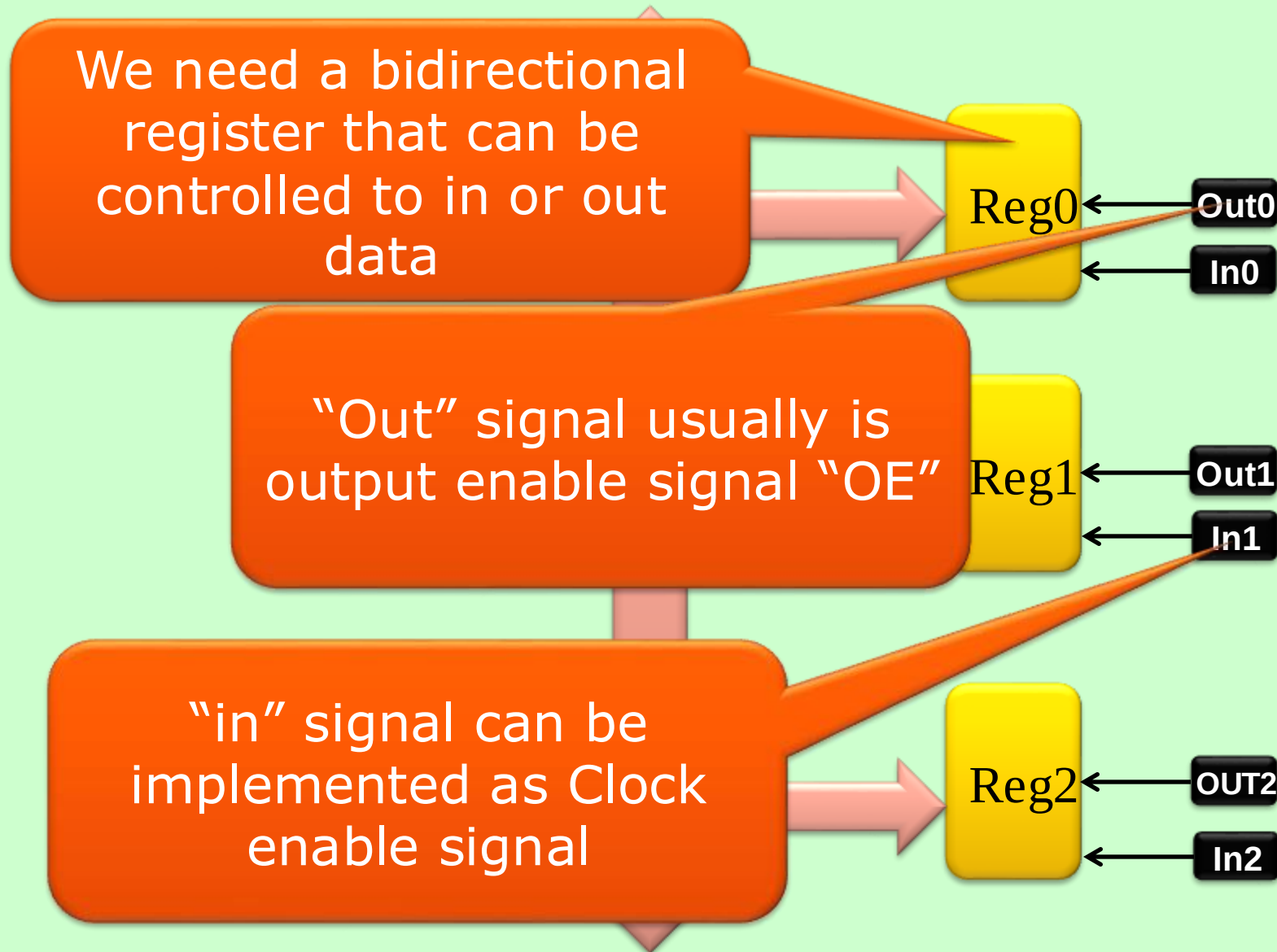
Reg0 ← Out0
← In0

"Out" signal usually is output enable signal "OE"

Reg1 ← Out1
← In1

"in" signal can be implemented as Clock enable signal

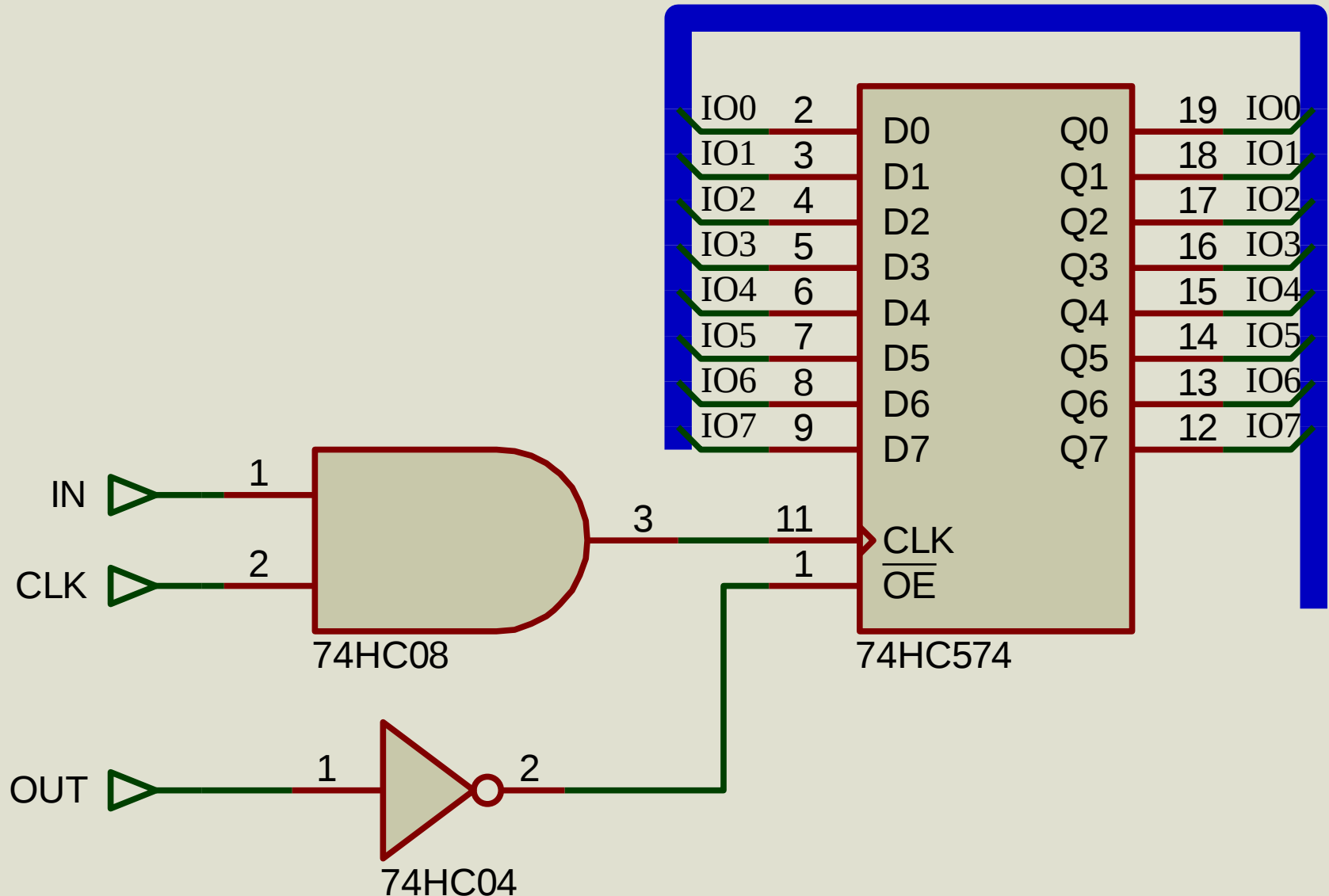
Reg2 ← Out2
← In2



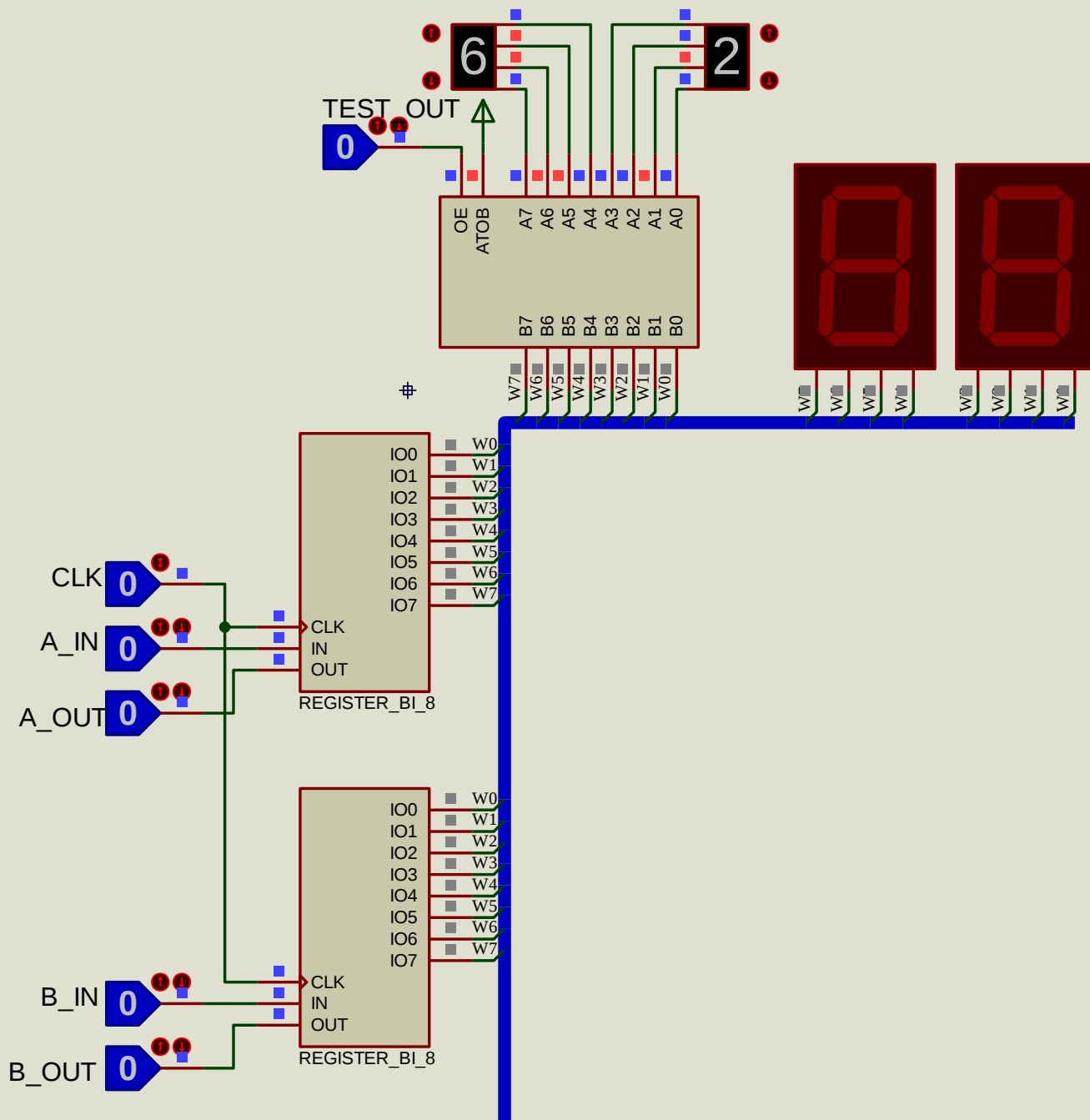
Registers Used in CPU –IN-OUT

- Registers used in CPU design must have the ability to transfer data to/from other units
- This feature is called Register Transfer
- Each Register can do either OUT or IN operation
- IN operation is equivalent to Write Data
- OUT operation is equivalent to Read Data
- A typical implementation of this register consists of
 - D-type FF and
 - Tri-state buffer

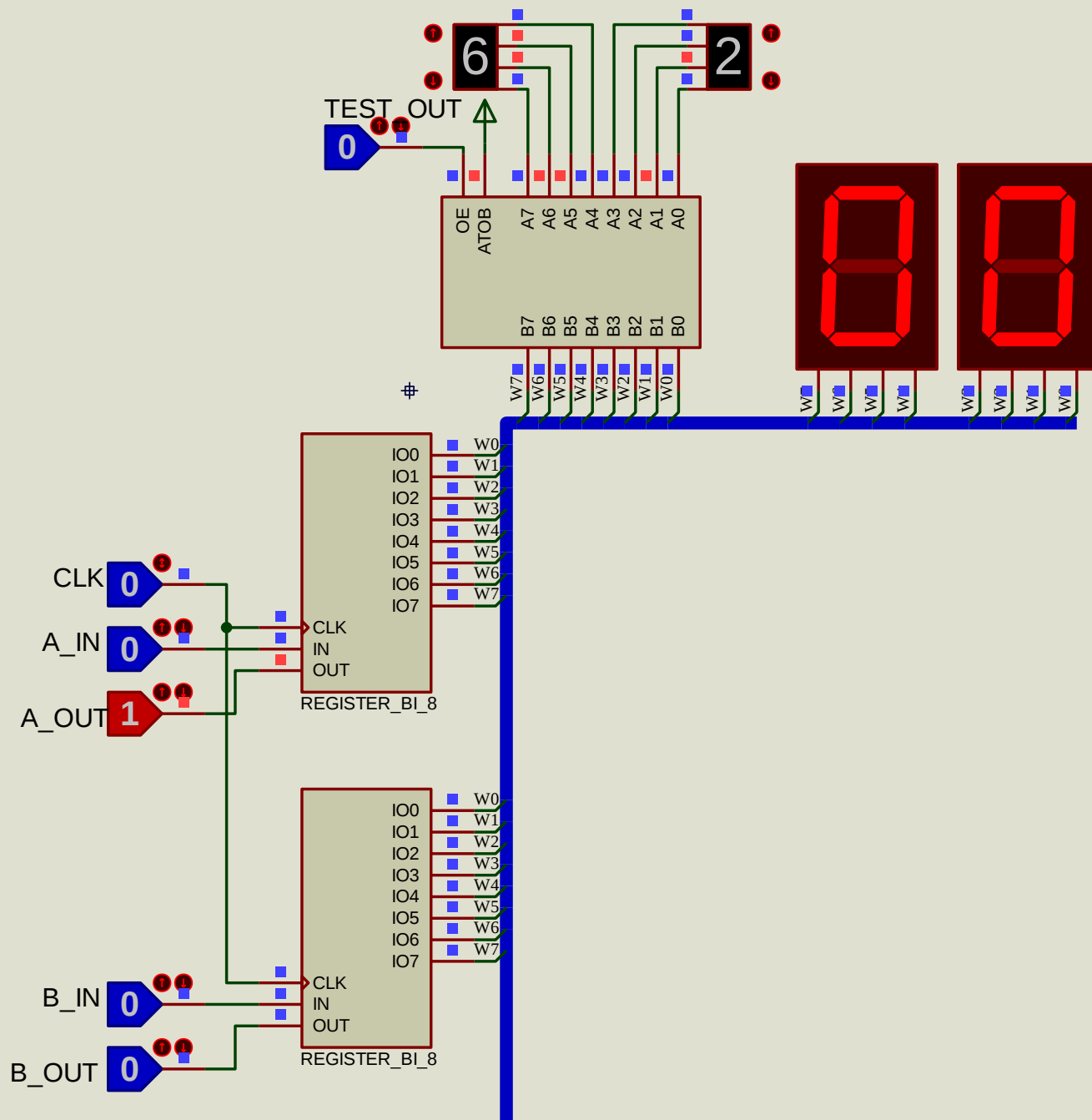
CPU Register Implementation



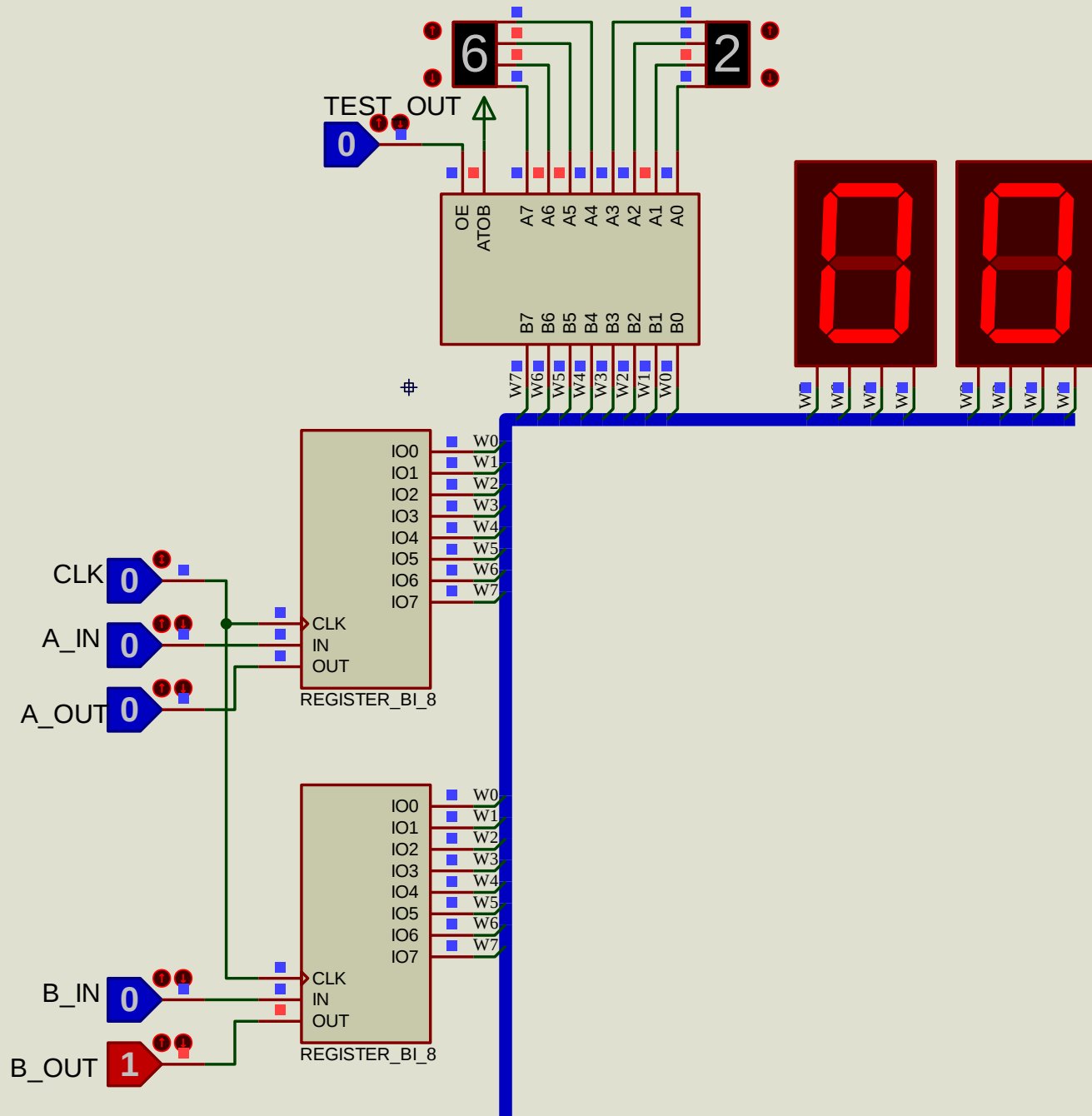
Bus



Bus

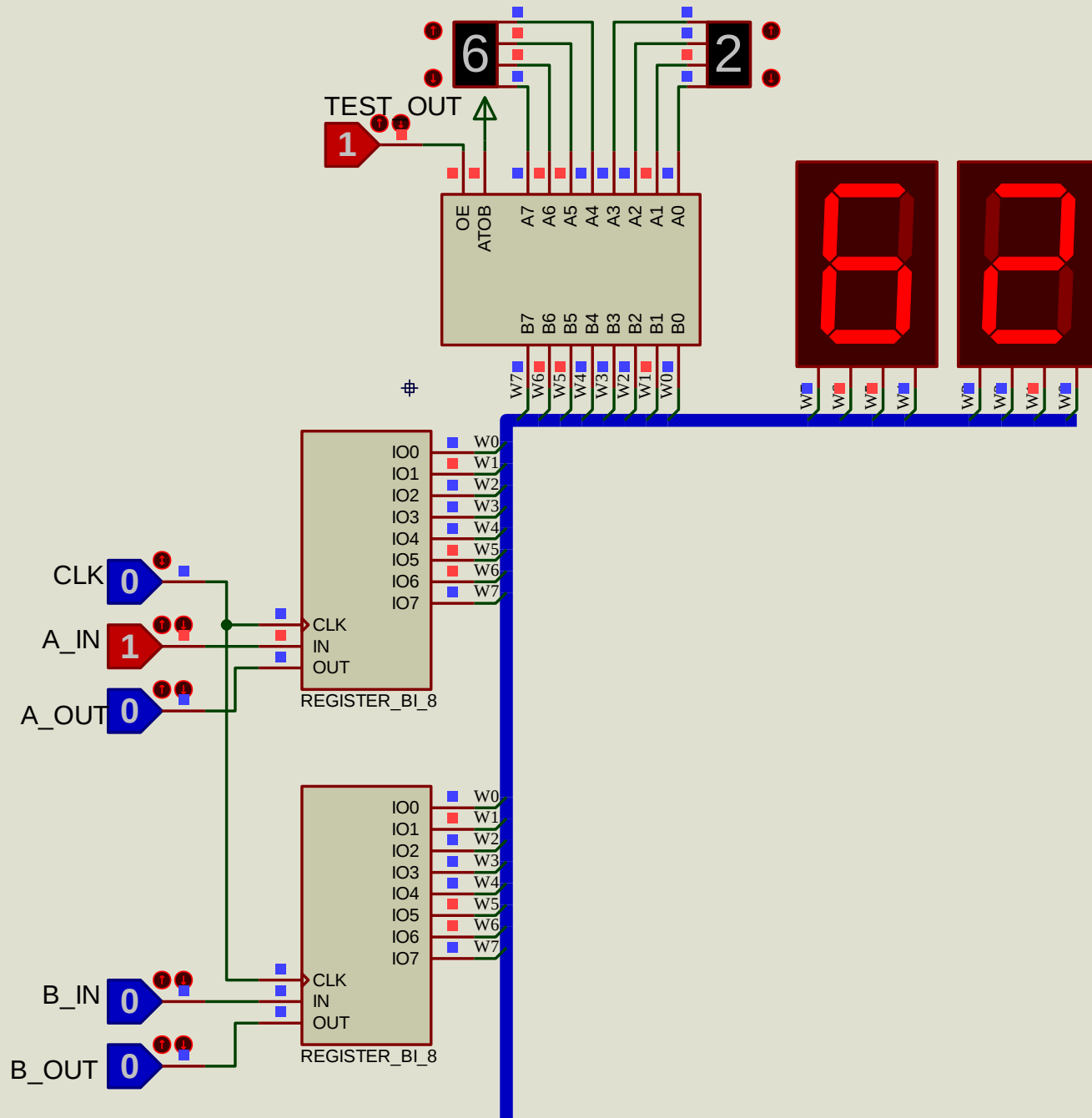


Bus

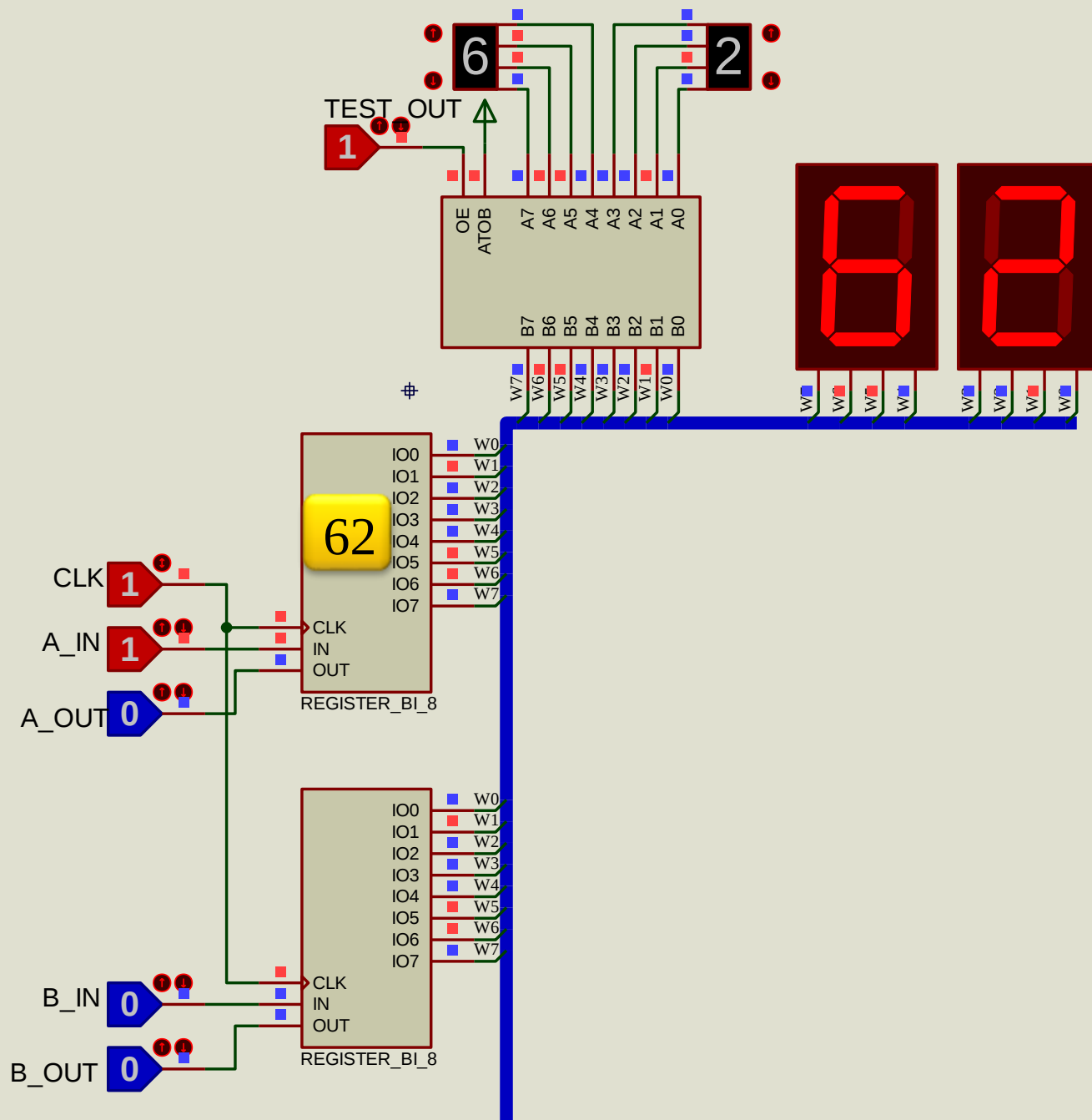


[illegible]

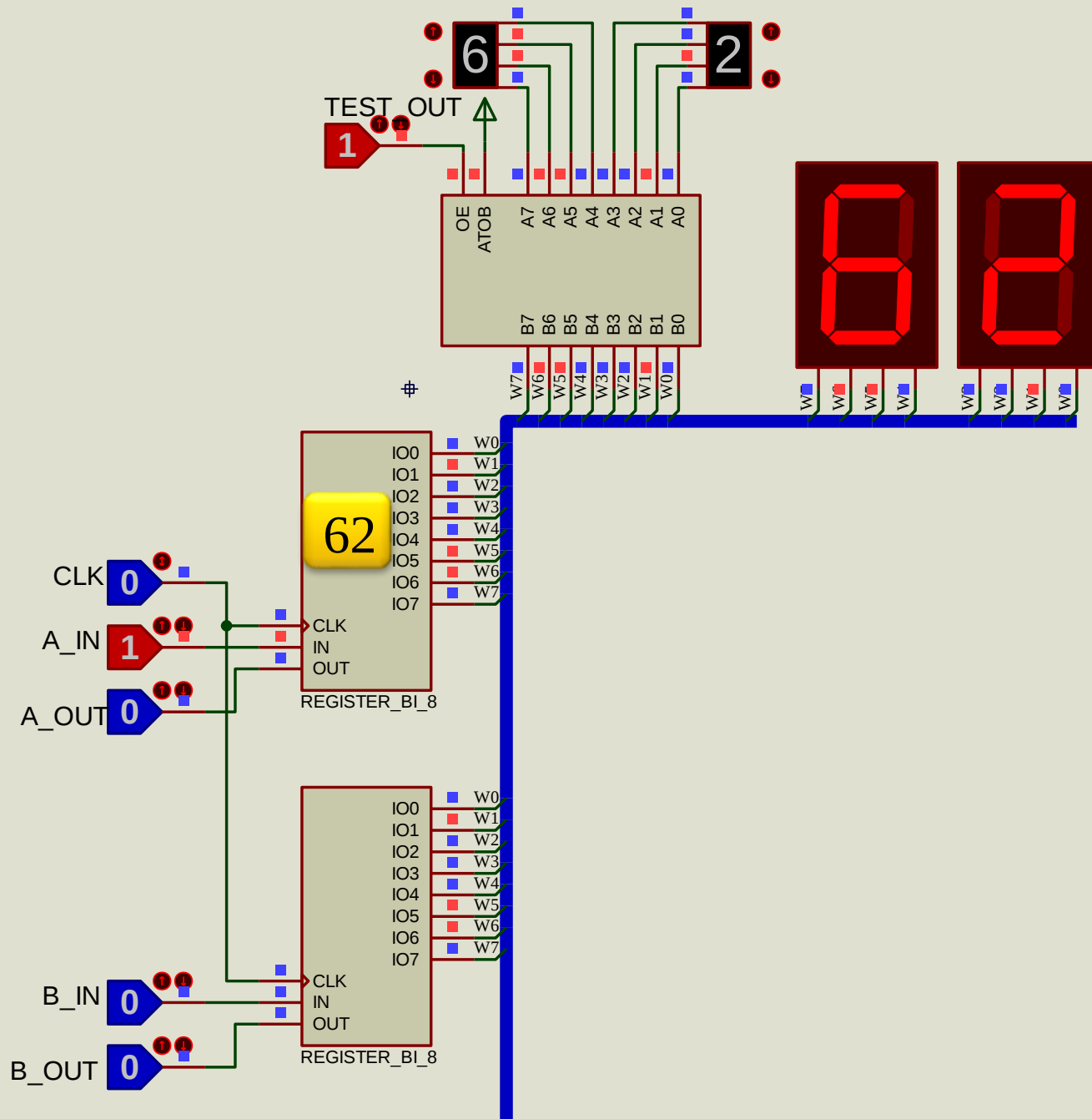
Bus



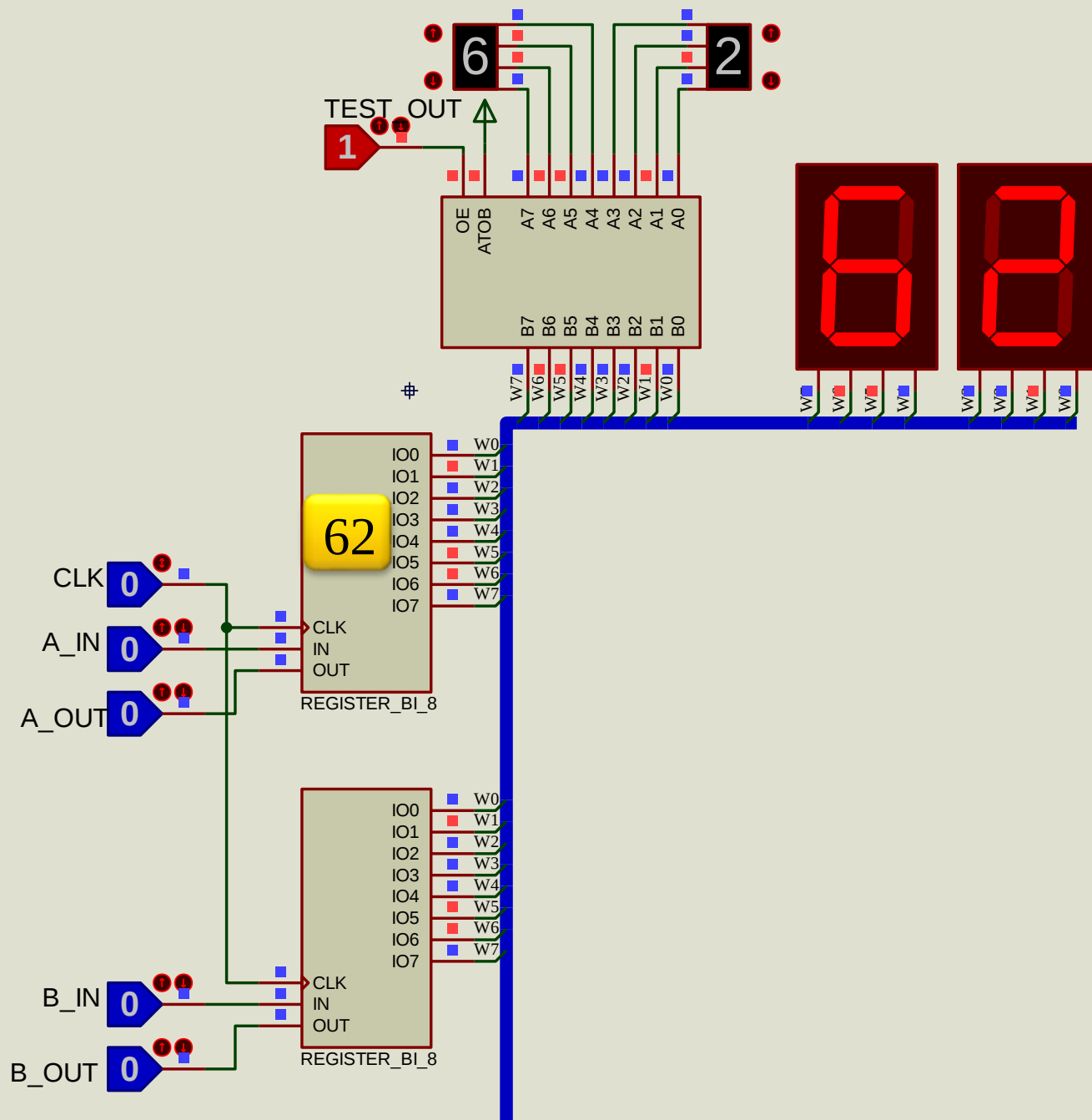
Bus



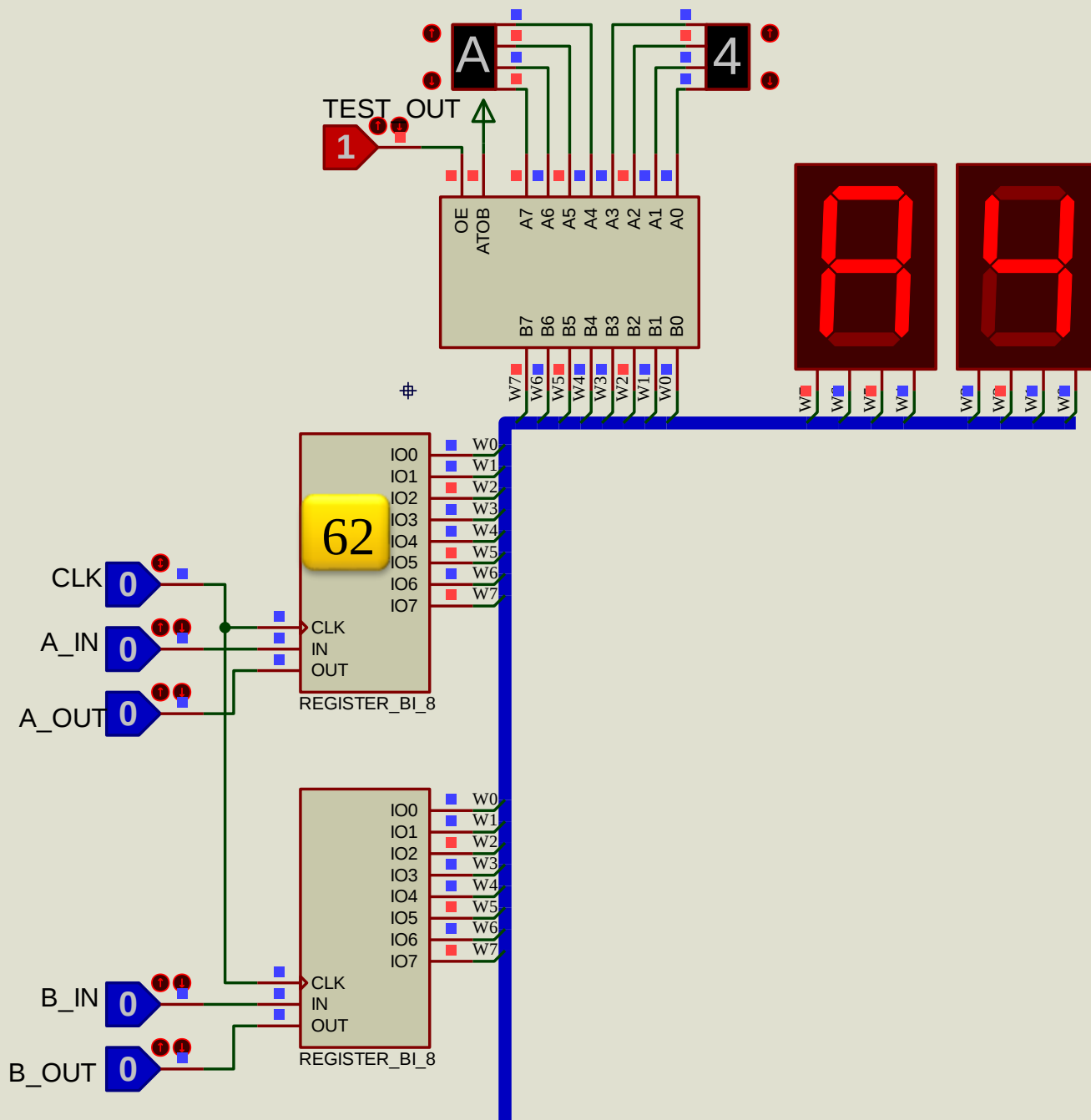
Bus



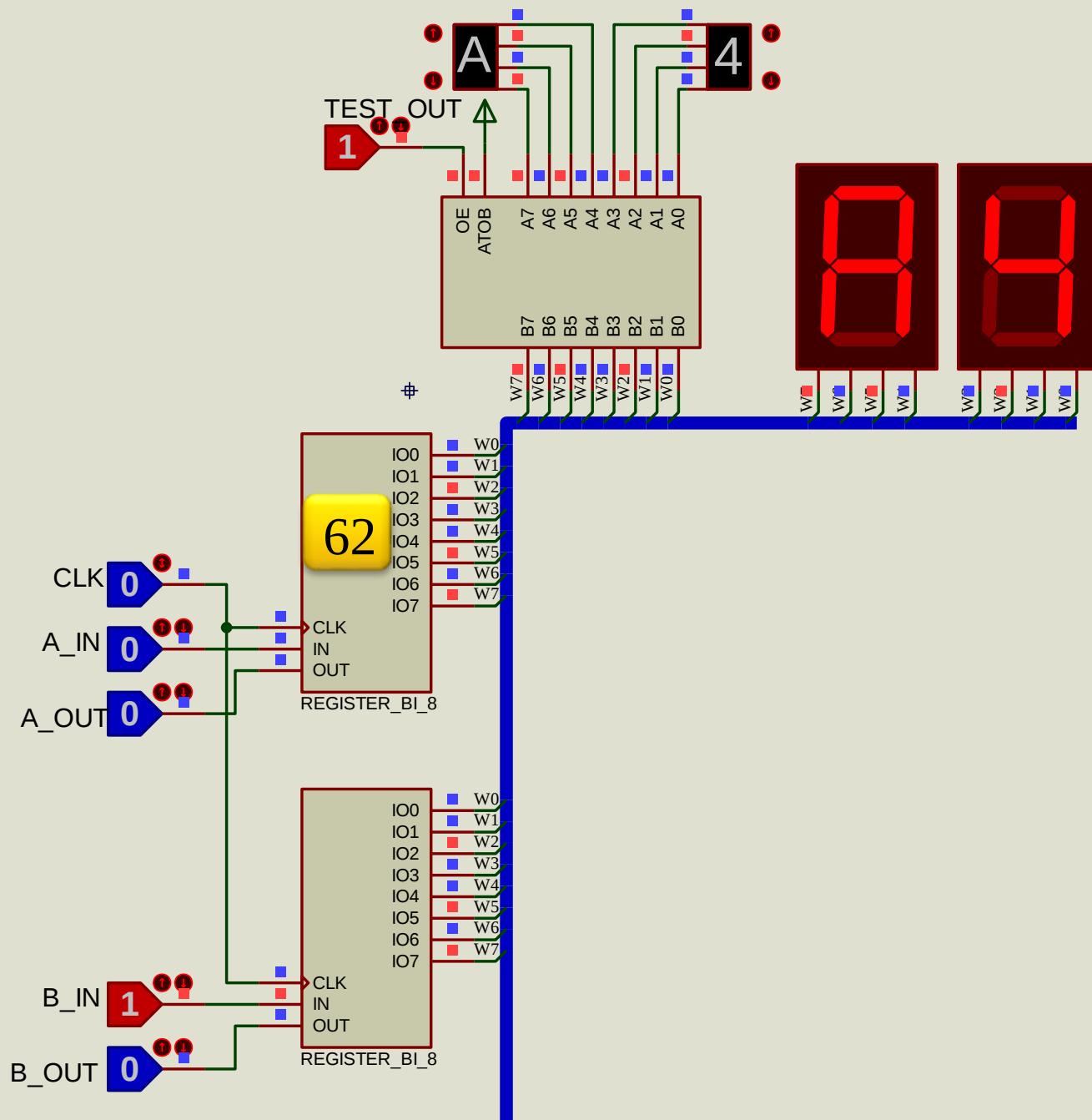
Bus



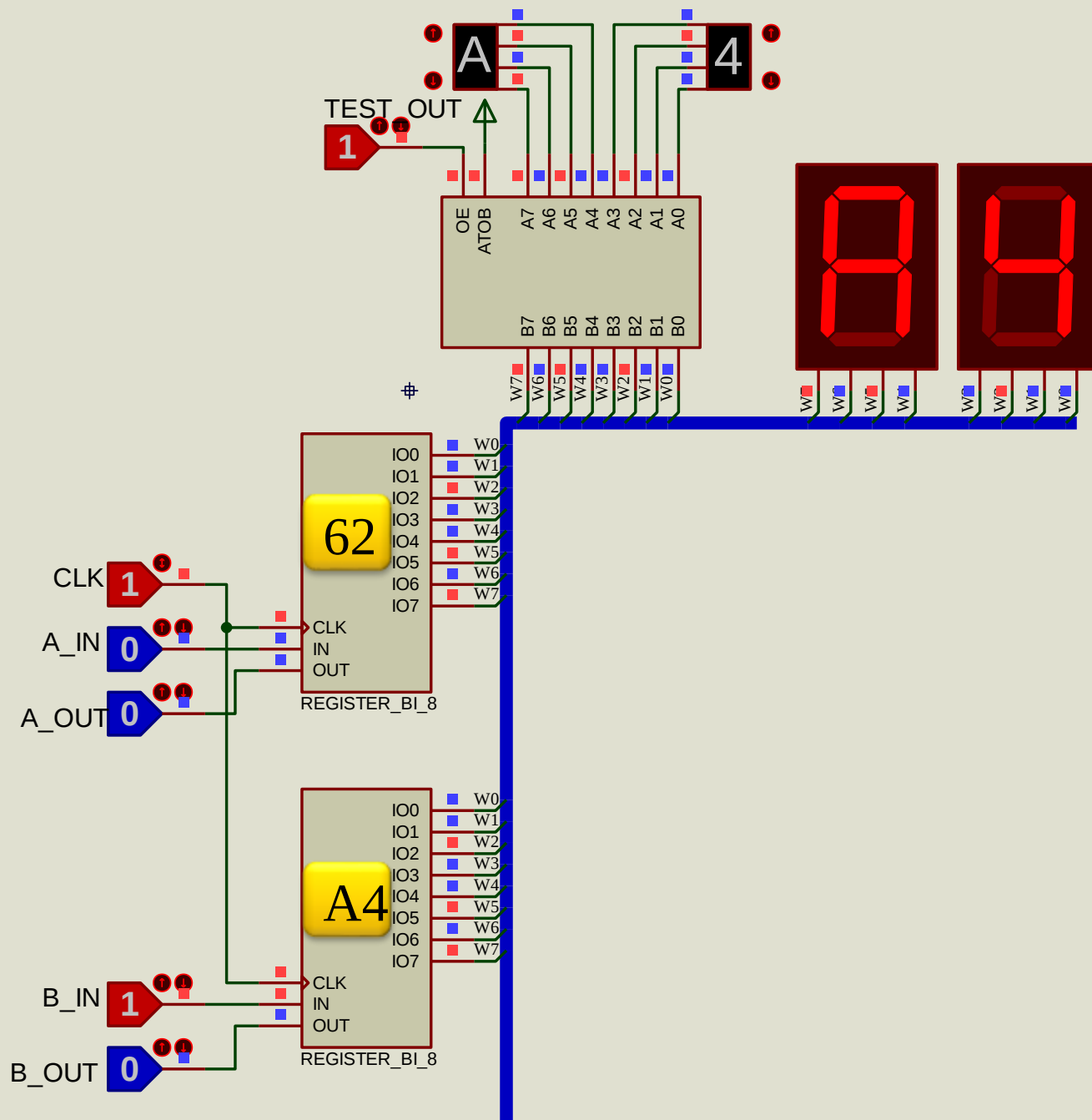
Bus



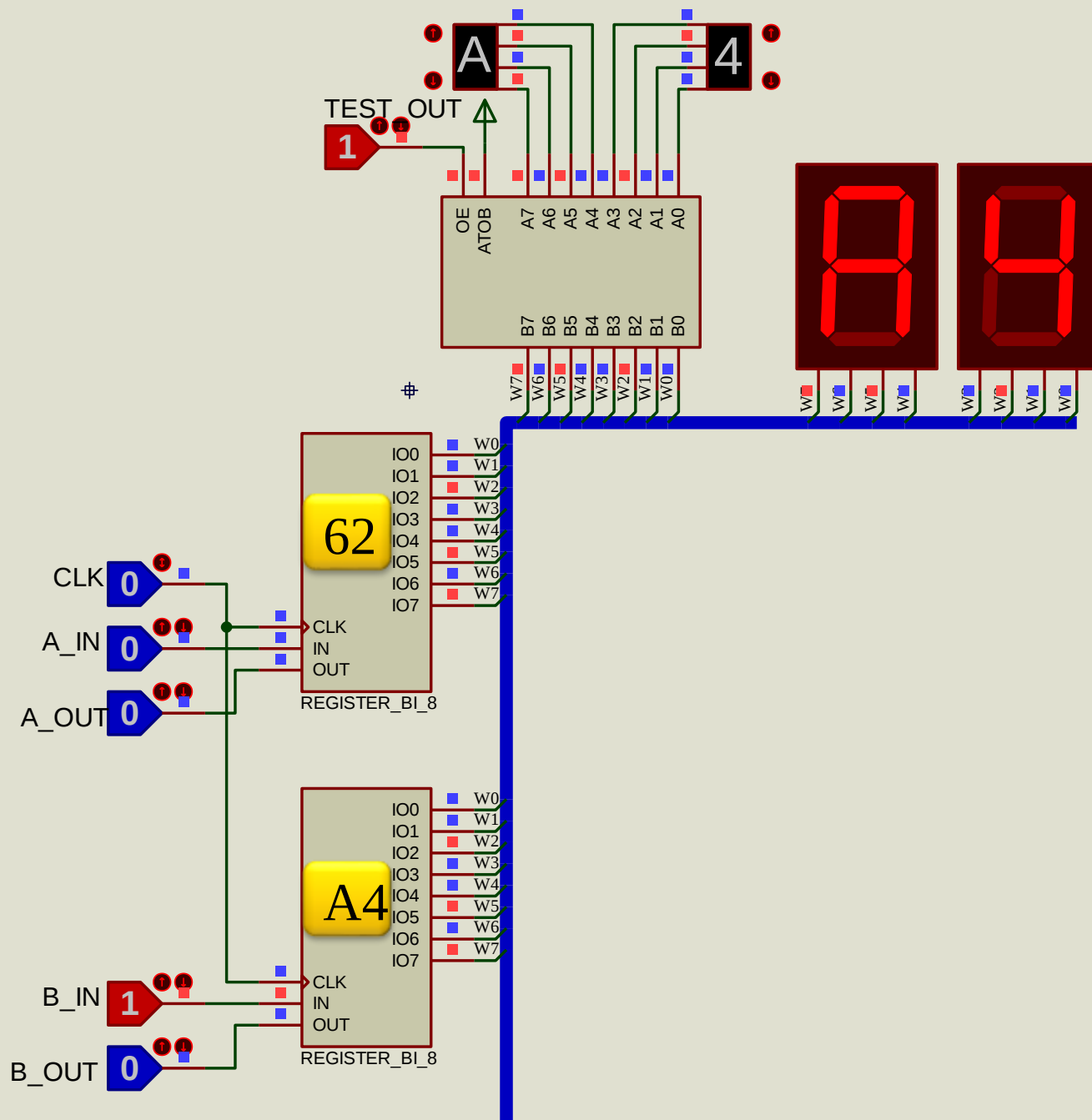
Bus



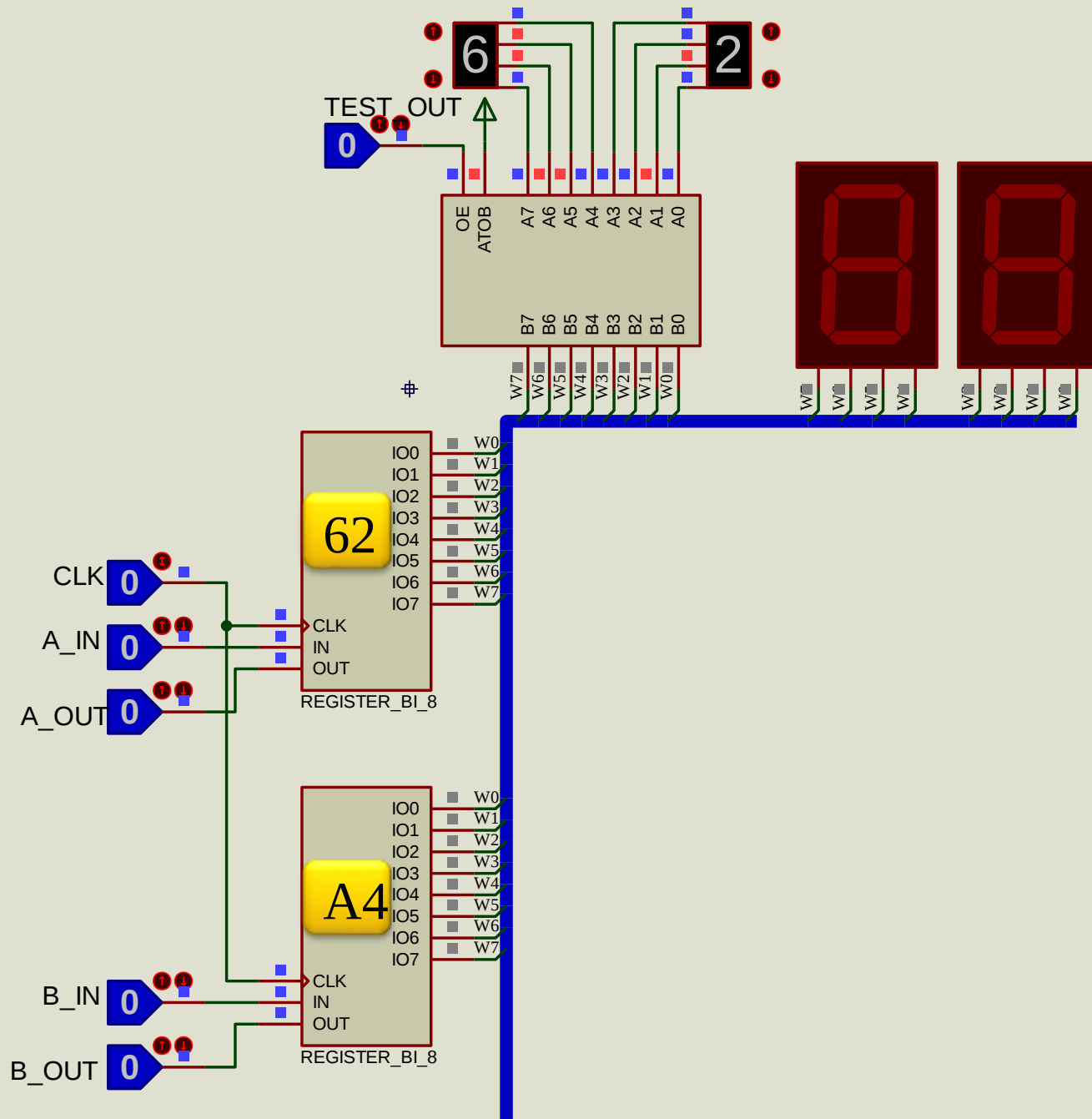
Bus



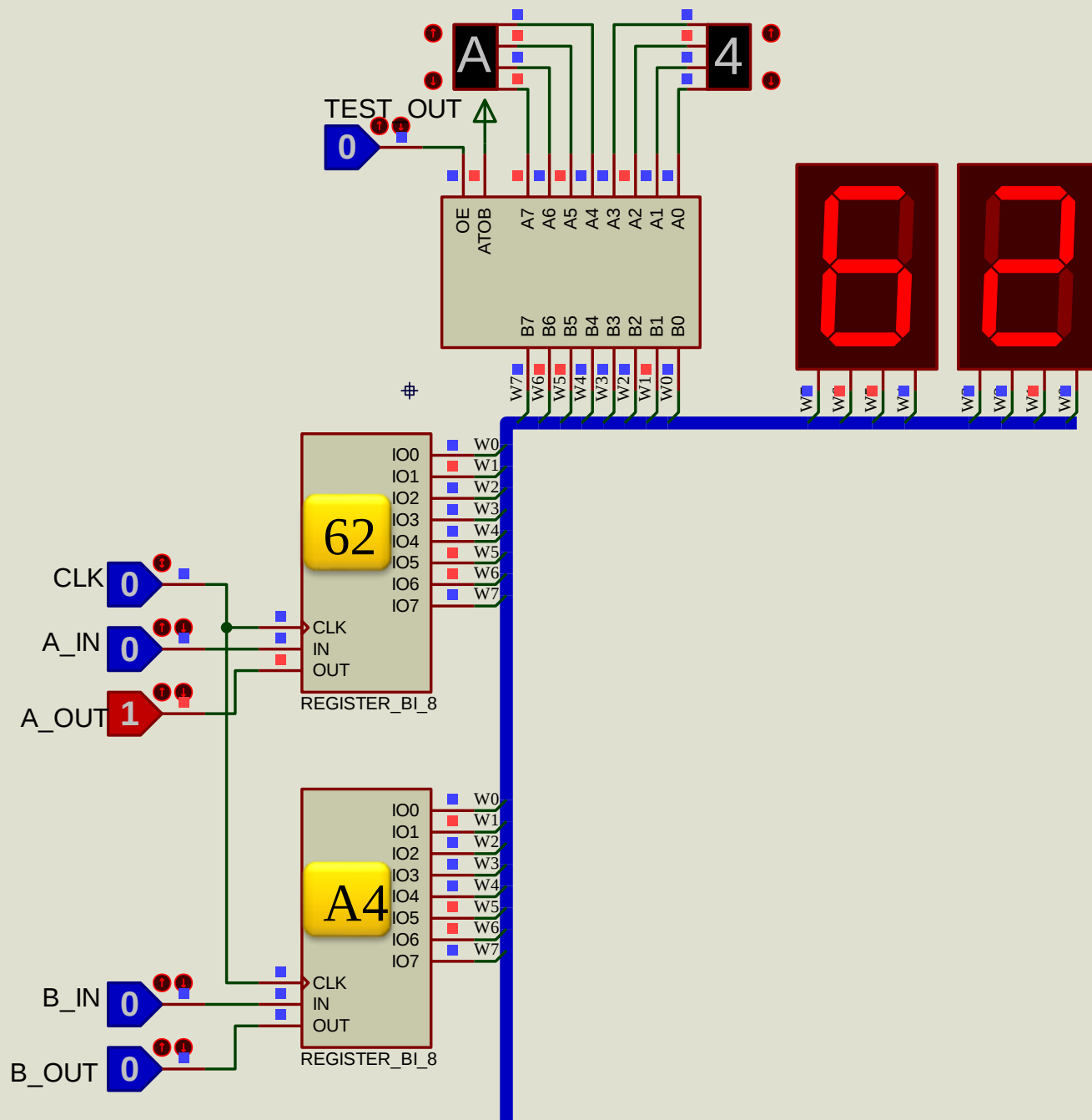
Bus



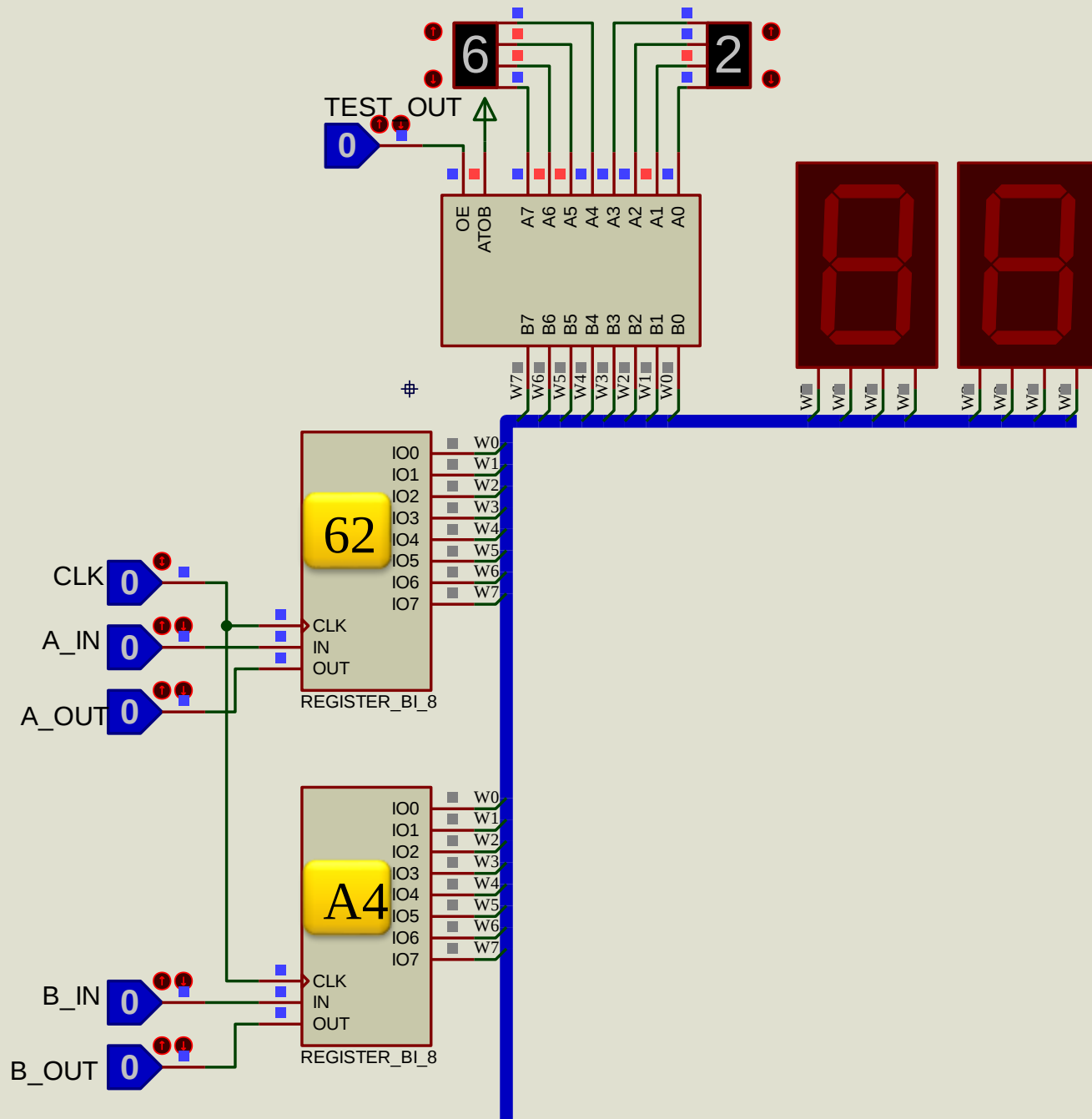
Bus



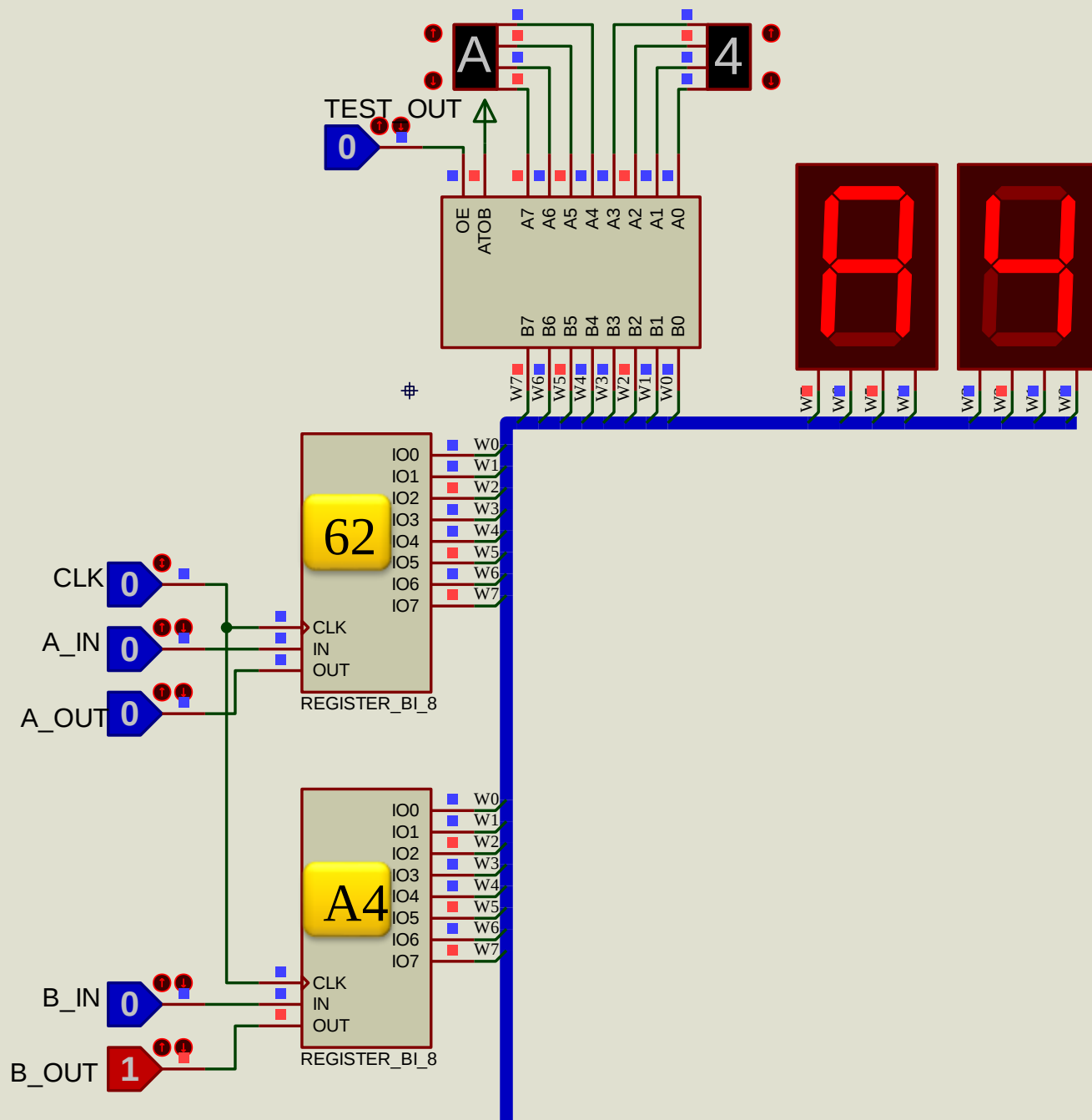
Bus



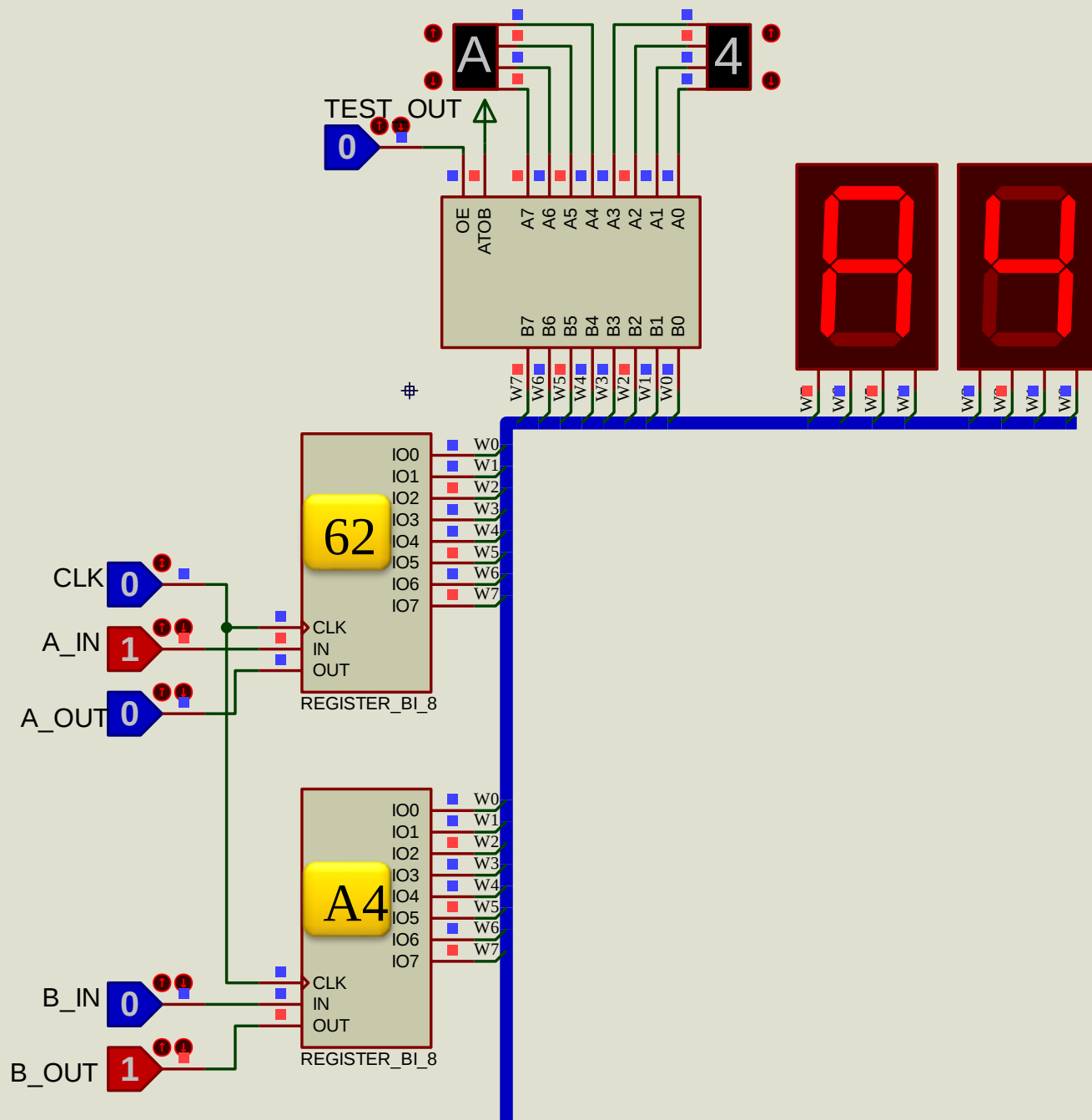
Bus



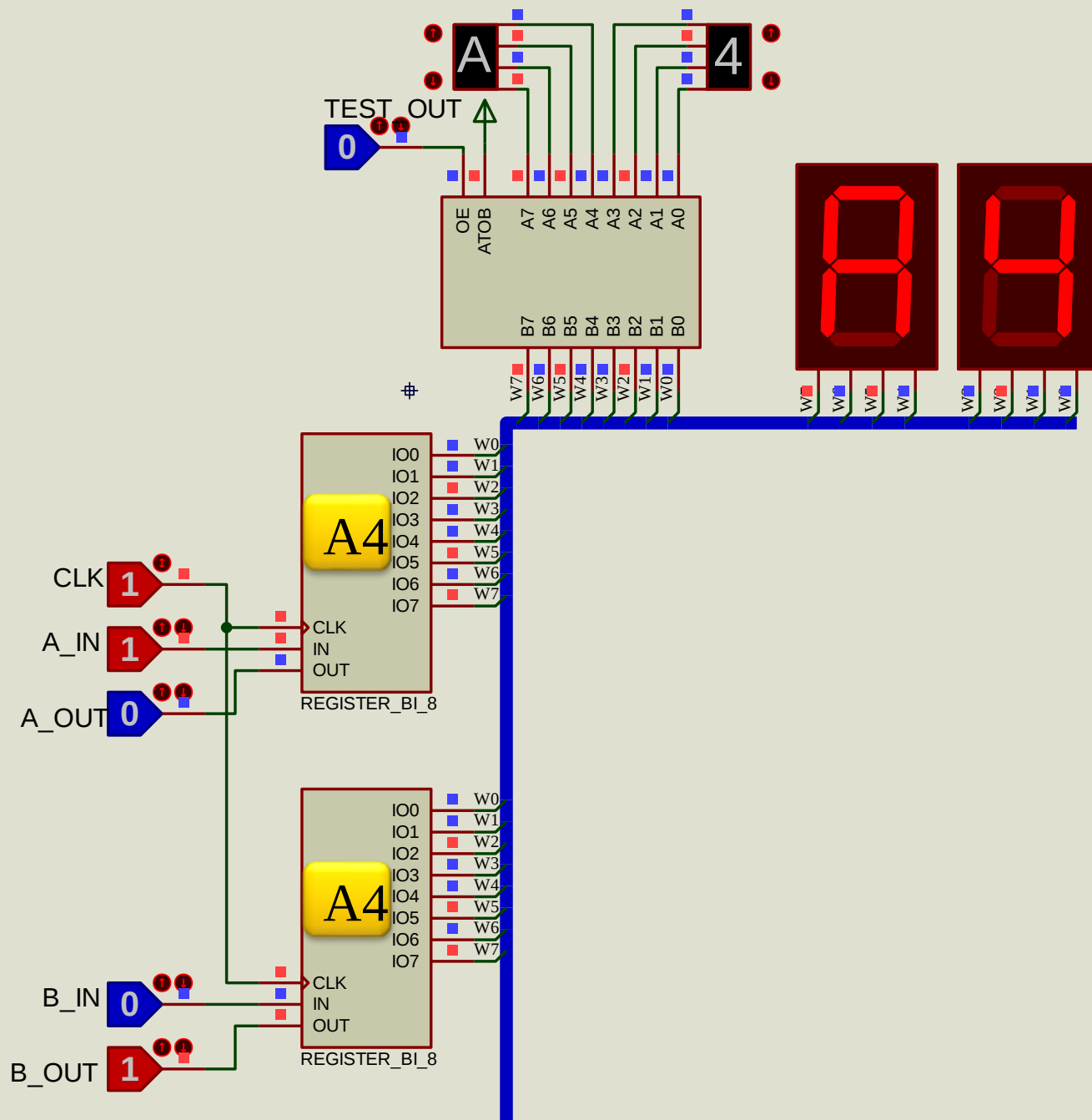
Bus



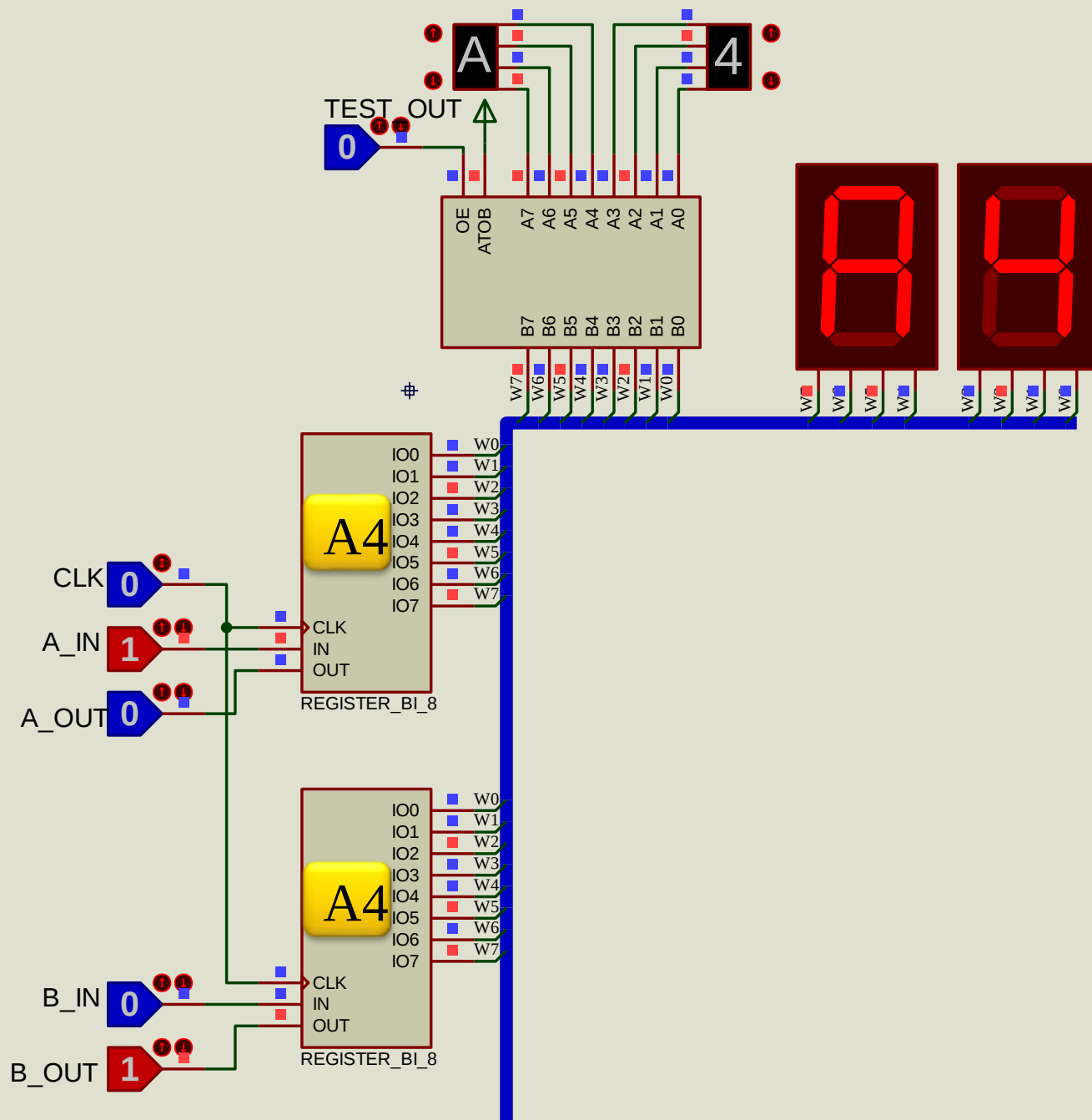
Bus

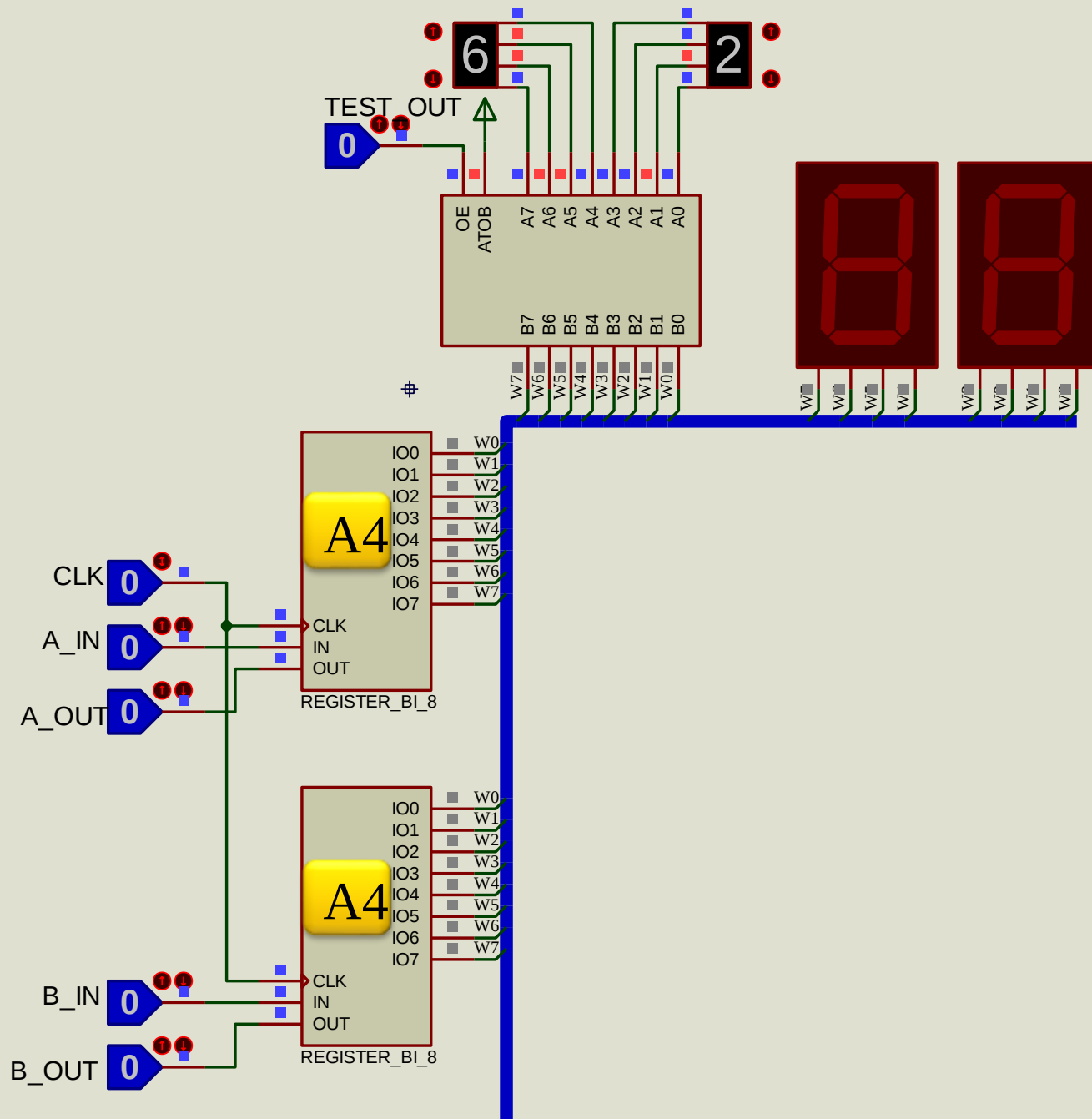


Bus



Bus





Bus

